Design • Build • Program • Learn • Compete

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

February 2013

Build The KronosFlyer

Final Assembly And Setup

Ready To Fly!

✦ Broadcasting Video With Your Robot

Add video to your bot, plus overlay text data for feedback, as well.

✦ Then And Now

Robot Telepresence And Personal Assistants

✦ Mr. Roboto

- **Can Cheap ESCs Do The Trick?**
- **Using The Digilent chipKIT MX3cK**

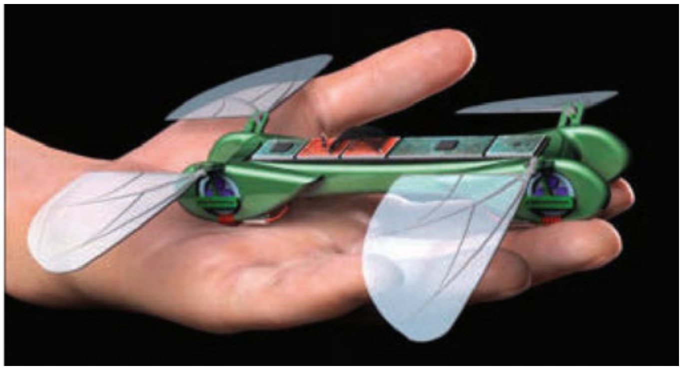
New Mobile App! SERVO Goes Where You Go

Available For:
iOS - Android
Kindle Fire

SERVO MAGAZINE

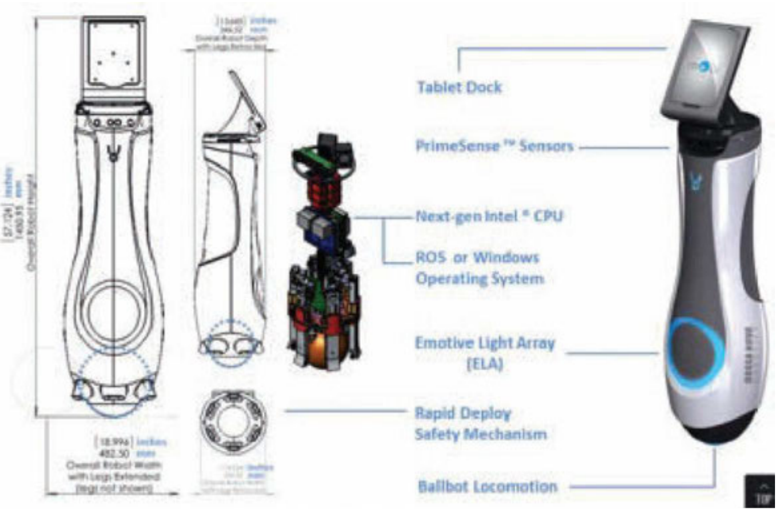
02.2013
VOL. 11 NO. 2

Columns

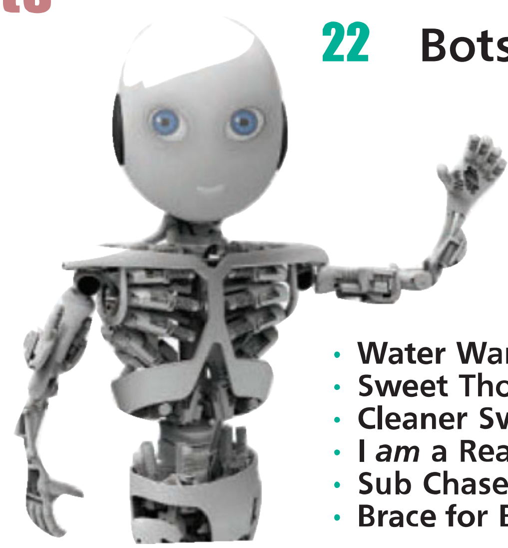


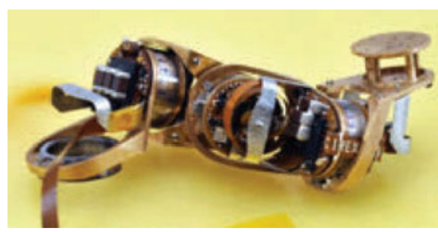
PAGE 08

PAGE 74



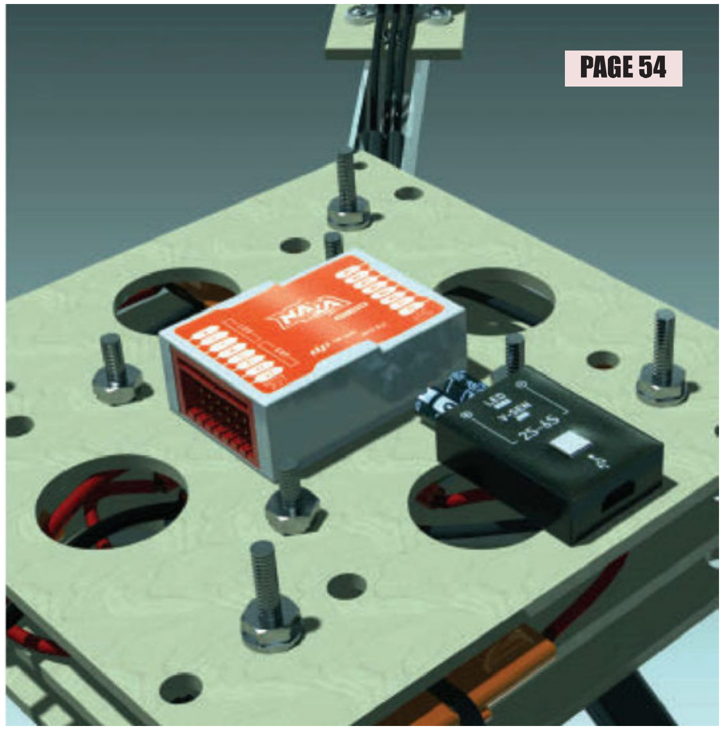
Departments

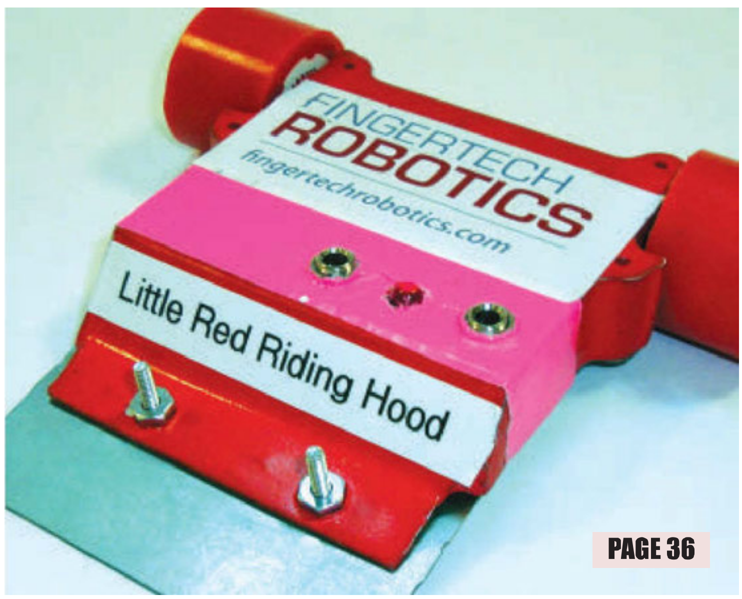




SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...





The Combat Zone...

Mind / Iron

by Bryan Bergeron, Editor

The Joy of Prototyping

There's nothing like prototyping to get the creative juices flowing. Sure, you can work out a robot design on paper or in a 3D CAD program, but there's something missing. Perhaps it's the tactile feedback, the ability to instantly and instinctively unplug one component and substitute another. Or, the ability to perform 'illegal' operations, such as mangling a connector to make it mate with a device not made to spec. It's hard to innovate — that is, create something new — if it has to first be defined in a software package. For example, I have yet to see the equivalent of gaffer's tape in a robot simulator or CAD program.

Sometimes, a little bubblegum and string can work wonders. As shown in the accompanying **photo** to the right, I'm developing a prototype of a sound tracking robot, based on both amplitude differences and time delay. What looks like a tangled mess is actually an orderly mess, centered on an Arduino Uno with Grove shield. The shield is inexpensive and accepts a variety of Grove cables with servo, power, or four-pin connectors. The cables are a joy to work with — as soon as you cut off the locking mechanism on the four-pin connectors that makes them impossible to dislodge.

At the end of one cable is a 567-based phase locked loop (PLL) board from Ramsey Electronics. I modified the board with the addition of a signal diode and bypass capacitor so that it would drive a digital pin on the Arduino. Then there's the artificial ear — the inverted red silicone cup with a modified Grove microphone sensor mounted on the back. I replaced the original Grove omni-directional microphone element with an inexpensive directional electret element from Digi-Key. Then, there are the jumpers between cables made with Futaba-compatible servo extender connectors.

You might also notice the black gaffer's tape holding down the LED indicator buttons and the lead to the servo. In the case of the LEDs, the tape keeps the fingernail-sized LED boards pointed upwards where I can see them. As far as the servo lead goes, the tape is an ounce of prevention. For some reason the folks at Grove decided to bring power to the servo through a male connector — meaning the Arduino's 5 VDC and GND connections are exposed. With all those cables and end connections, exposed power sources aren't a good thing.

This is just one of many possible prototyping environments. It happens to be my favorite for working with Arduinos, but when I have an analog circuit based on a few chips, I use a standard breadboard environment. If it's a super-simple circuit, I might even pull out the wire wrap tool and wire components together in a few minutes.

Rarely do I heat up the soldering iron for a prototype. Soldering irons are for production, when I know I won't be changing a component or connection every 10 minutes — with one exception: Tubes.

Tubes are in a world of their own. Because of the high voltages involved, I don't use breadboards. What I do use is an old guitar amplifier chassis with clean sockets, rows of solder posts, and a hefty high voltage and filament power supply. The other major feature is a set of circuit breakers on all the supply leads.

What's your prototyping environment of choice? Do you have something that you've built that's worth sharing with readers? If so, please drop me a line. **SV**

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, Kevin Berry, Dennis Clark, R. Steven Rainwater, Michael Simpson, Gordon McComb, Daniel Ramirez, Dave Graham, Mike Jeffries, Pete Smith

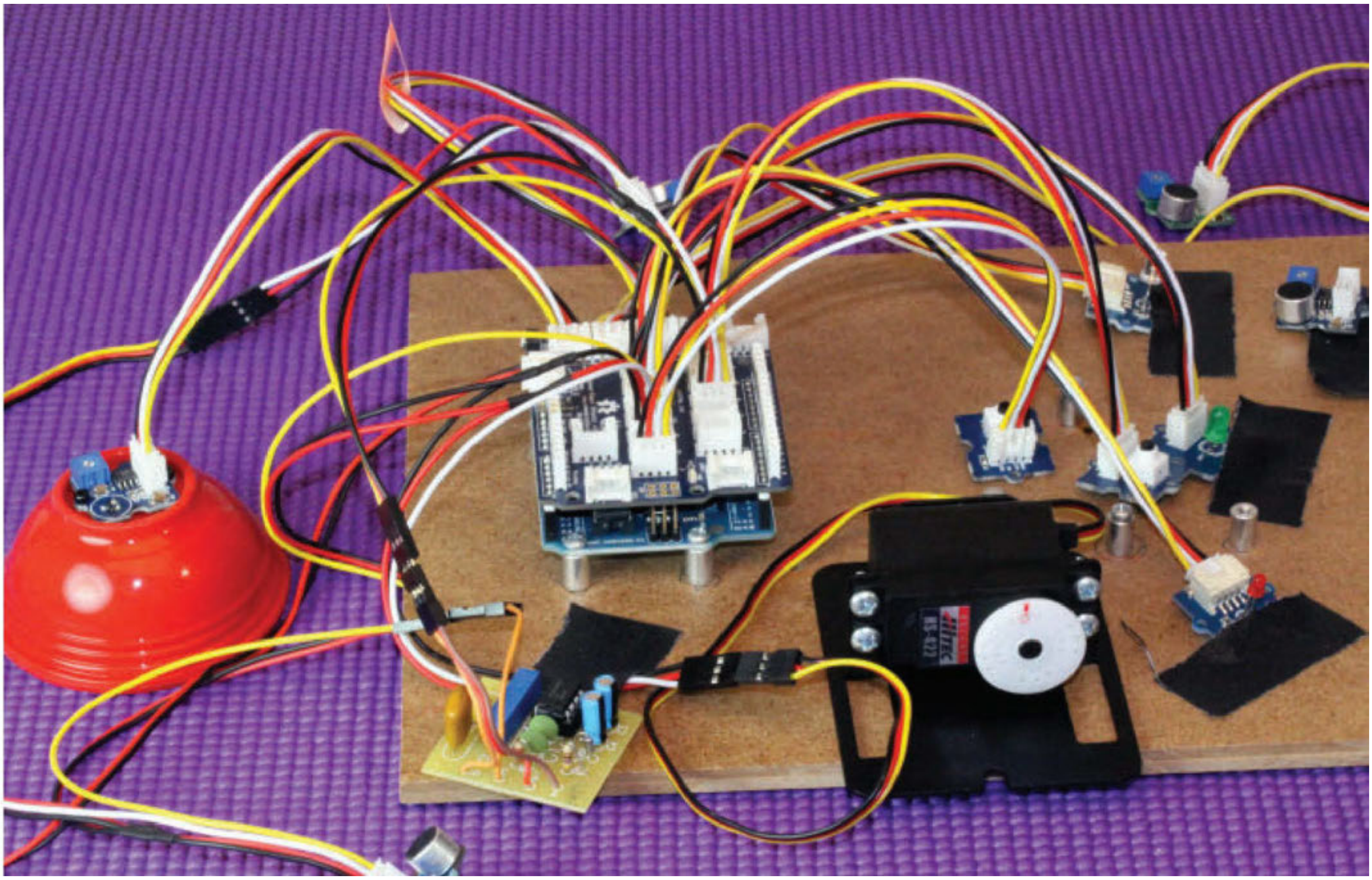
CIRCULATION DEPARTMENT
subscribe@servomagazine.com

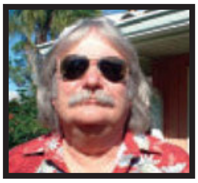
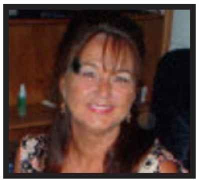
MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Sean Lemieux

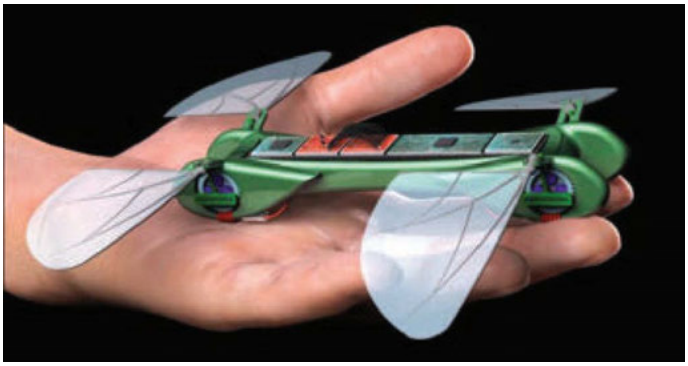


Robytes

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com.

by Jeff and Jenn Eckert



TechJect's Dragonfly flies like a bird, hovers like an insect, and relays snoop data.

UAV for Everyone

If you have been worried that aerial spybots might someday become cheap enough for deployment by your neighborhood peeping Tom, take note: Someday has arrived, along with the Dragonfly UAV from TechJect (**www.techject.com**). The spinoff from Georgia Tech's Robotics and Intelligent Machines Department was obviously inspired by the abilities of the real dragonfly, which the team describes as "the king of them all ... the top of its food chain and the fiercest predator in the sky, with unmatched flight performance in the insect world."

The Dragonfly — developed with $1 million of funding from the US Air Force Office of Scientific Research — is available with a choice of three different electronics packages, ranging from basic flight control using gyroscopes and accelerometers, to more advanced packages that include GPS, HD video, multiple cameras, and wireless communication (2 Mbps data relay). The packages are said to be the lightest in the world, ranging from 3 to 5 g.

According to TechJect, "With 20 environmental sensors, cameras, and GPS capabilities, it can be developed to track athletes, outdoor events, or be integrated with an app as a fun way to plot or retrace a day of skiing, hiking, or whatever you like."

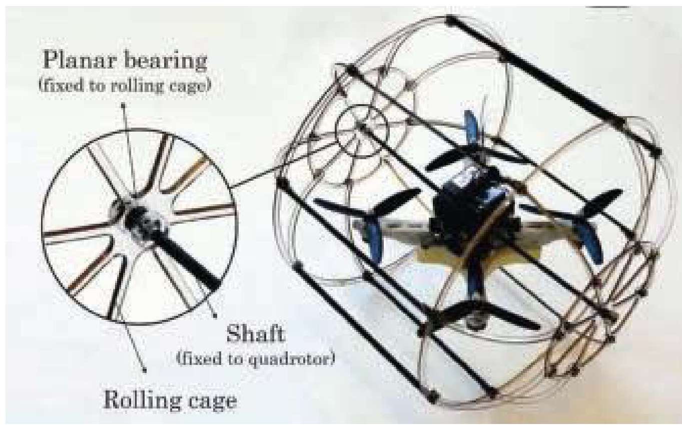
No assembly is required, and it can be flown R/C or controlled with an iPad, smartphone, or computer. The LiPo cell provides 10 to 30 min endurance, depending on flight mode.

Getting back to the "cheap enough" factor, a Dragonfly is expected to cost only about $250 when they hit the market — perhaps by the time you read this. An overly slick but interesting five-minute video is accessible on the company website.

Monopoly for Bots and Meatbags

If you're tired of the standard game of Monopoly (and who isn't?), you might want to check out the Futurama version, inspired by Matt Groening and David X. Cohen's TV show of the same name. Prominently featured, of course, is Bender Bending Rodriguez (or better known as just Bender) — the human-hating robot described in the series as an "alcoholic, whore-mongering, chain-smoking gambler."

Noted as being Monopoly for both "meatbags



The HyTAQ bot -- developed at Illinois Tech -- offers hybrid locomotion.

Flies and a Roll

Now and then you see something that makes you slap yourself in the forehead and ask, "Why didn't I think of that?" A good candidate is the Hybrid Terrestrial and Aerial Quadrotor (HyTAQ), developed at the Illinois Institute of Technology (**www.iit.edu**). We all know that rolling bots encounter obstacles and flying bots use a lot of power to stay aloft. Someone has finally put the two together to achieve the best of both worlds.

HyTAQ is basically just a garden variety quadrotor in a rolling cage. Simple, but pretty clever. While in terrestrial operation, the rotors push the unit along the ground. When something gets in its way, it simply flies over it. Experiments have shown that the unit can travel four times farther and operate nearly six times as long as an aerial-only system.

and robots," it features six custom tokens, new styles of houses and hotels, and 60 minute play rules for those who get tired of five-hour games (and who doesn't?). In the Futurama version, you can "hobnob with huggable industrialist Mom, dance with the Robot Devil, and take a mutated plunge into Sewer City and try not to get mutated while there." Available from various websites, the standard edition will run you in the neighborhood of $35. For a couple bucks more, you can get the special collector's edition which features a special gold Bender token.

Game board from the Futurama version of Monopoly.

The Future of Burgers?

If you're a recent college graduate who has been unable to find a job in your chosen field and you're planning on an alternative career at McDonald's, we have some bad news for you. If you're looking for better burgers at a better price, though, you're in luck. According to the folks at Momentum Machines (**momentum machines.com**), the average fast food restaurant spends $135,000 every year on labor to produce hamburgers, which totals $9 billion for the industry as a whole. Momentum intends to change that via (what else?) robotics. The company's "alpha" machine — also known as "The Burgeon" — replaces all of the hamburger line cooks in a restaurant in a 24 sq ft package that actually grinds the meat, forms the patties, slices the toppings, cooks the burger, and assembles the finished product to the tune of 360 burgers per hour. In addition, a soon-to-be-introduced version will offer the ability to provide custom grinds (e.g., 1/3 pork and 2/3 bison) for every customer.

We briefly mentioned these "Pattyburgers" in last month's magazine, however, the website still doesn't provide specifics about price, but the company claims that a machine can pay for itself in the first year of operation. It also doesn't mention how or where to buy one, nor does it provide any kind of video evidence that it really works.

According to an article from San Francisco's Mercury News, a prototype sits in a warehouse in the city's South of Market (SoMa) neighborhood. Plus, Momentum is funded by "hardware incubator" Lemnos Labs (**lemnoslabs.com**), so we're inclined to believe that the machine isn't just vaporware. At some unspecified time, the company "will launch the first restaurant chain that profitably sells gourmet hamburgers at fast food prices."

Momentum's Burgeon creates up to six a custom burgers per minute.

Flipper Headed for Retirement

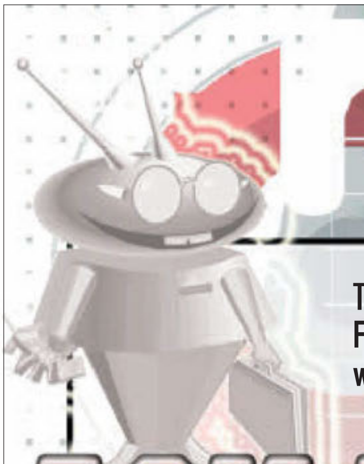
The employment outlook is also looking grim for dolphins, sea lions, killer whales, and other animals that work for the US Navy's Marine Mammal Program, which is a part of the Space and Naval Warfare Systems Command (SPAWAR, **www.public.navy.mil/spawar**).

The program has existed since the 1960s, but was not declassified until the early 1990s. Although there was no truth to rumors that the Navy was training a cadre of killer dolphins, it is true that various friendly and trainable species have been used for sniffing out and marking mines, and a variety of other nonlethal functions.

According to Capt. Frank Linkous, head of the Navy's Mine Warfare Branch, the program will be phased out by 2017. Folks who have opposed the use of sea mammals for such purposes will be elated but, in fact, the change is motivated primarily by economics. It is very expensive to transport, house, and otherwise maintain this huge aquatic zoo.

One of several replacements will be the Knifefish, a 7 m (20 ft) UUV (Unmanned Undersea Vehicle) that should be deployed just in time (see the August 2012 issue for details). Also to be deployed are some Kingfish UUVs and the German-built SeaFox mine hunters. **SV**

A Navy dolphin carries a mine marker to its destination.



Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by
Dennis Clark

This month marks four years of Mr. Roboto for me. Much has changed since I started. For one thing, there are more robot controller options out there that have a lot of power and aren't very expensive; for example, the Digilent chipKIT series and Raspberry Pi to name just two. There are new platforms like Robot OS (ROS) and new "glue" programs like RoboRealm. Pololu and SparkFun are carrying newer and better motor and servo controllers, and IMU (Inertial Measurement Unit) devices have become affordable to the robot hobbyist for aircraft and walking robots.

Robo-One style walking robot events have unseated mini Sumo as the totally cool robot competition to compete in. If you haven't seen the movie "Real Steel" yet, do so; I believe we are only a few years away from that level of full-sized humanoid robot competition. In my opinion, our only hold-up is the self-contained power system required to just get a walking robot to move!

It has been exciting and fun. Here's to even better times ahead! Hackers and hobbyists, if you have questions, suggestions, and even examples to share with us, please let me know and I'll help and/or publicize it. I just love this hobby!

First off, I have to apologize that my "reach exceeded my grasp" this month, and I was unable to complete the project required to get an Arduino to control the Roomba robot. I had issues with timing between the Arduino and the Roomba that I could not explain. At first, I was convinced that the Roomba was a 3.3V serial interface that was not reliable when connected to the 5V Arduino UART. However, Roomba documentation clearly states that the serial connection is a 5V UART. So, that idea didn't fly. I am working out these details now.

There were also some reports that the Arduino library for the Roomba was not robust enough, but I'm not convinced that it isn't the interface causing the problem, instead of the library. Finally, my nemesis -- the CABLE -- got in my way a bit as I sorted through the connectors to attach to the Arduino board. Grr. Some projects just don't work out according to plan!

Regardless, I will beat this project into submission and I WILL HAVE A WORKING SETUP next month. In the meantime, I'll answer some questions that I had postponed while working on this Roomba problem.

On with the questions!

Q. Hey Mr. Roboto! It's Corey, again. I have to say, I'm liking the articles following my question! The PS controller was actually something I had been looking into, so I was very glad to see you write about it.

I actually have another product question for you. I promise this one will be much simpler than the chipKIT one.

There seems to be a lot of articles talking about controlling motors in robots using the tinyESC by FingerTech Robotics. Although I have yet to personally use this, I would absolutely love to! The only thing that has kept me from buying any of these is the price since they are $35 a pop. In browsing the HobbyKing's site, I noticed that they had a bunch of ESCs (electronic speed controller), and that they are really cheap (as in $6-$15 for the 20A and below models). Could you do a comparison of these cheaper ESCs to the infamous tinyESC? I know all of the spec differences between them, but I want to know how they differ in real working situations. If they stack up nicely functionality-wise compared to the tinyESCs, I would love to buy five of the $6 ESCs for the price of one tinyESC! Thanks!

— Corey Hastings

A. The $35 motor controllers are more for robotics use or for when precise control over a motor is desired. These tend to be brushed DC motor controllers which are what we typically use on our robots. The HobbyKing ESCs — if you look closely — are for brushless motors, which I don't think you're using unless you are building an R/C aircraft.

So, the tinyESC units are for brushed DC motors. They are also designed for use with an R/C receiver rather than a robot controller. Your basic controller can still use them, but has to simulate the servo pulse given through the R/C receiver. For my money, I would use the Pololu Simple motor controller. It will take input via a variety of sources, including your microcontroller UART.

The Pololu controllers allow a huge amount of custom tuning — some of which you can do with the tinyESC and none of which you can do with the HobbyKing versions. If you don't need that level of control or want to go "on the cheap," then just get motor drivers and have your microcontroller control them directly. Chips and carriers for these are available at a variety of places. For little 1.5 amp or less motors, I use the good 'ol 754410 dual motor brushed motor driver chip. There are MANY options in this category! Hope that helps.

Q. I have a Digilent chipKIT MX3cK board that says it is MPIDE compatible like the UNO32 and MAX32, but I don't know how the Arduino ports match up to the MX3ck pins. Help me Obi wan Roboto! You're my only hope!

— Confused in Colorado, Charles

A. I talked to the good folks at Digilent about this, and they told me that a MPIDE map to the Cerebot boards (like the MX3cK) does not exist. The MX3cK does indeed have an *Arduino-esque* bootloader that allows it to be programmed through the MPIDE like the UNO32 and MAX32 though, so there should be a solution to this problem.

The first thing you have noticed is that the MX3cK has a different pinout. Because of that, the pin orders and locations will be different between the UNO32 and the MX3cK. The MX3cK has five PMOD connectors; each has power, ground, and signals. You can change a jumper to select between 5V and 3.3V to each PMOD connector separately. However, the PIC32 series is still a 3.3V part, and these boards are still 3.3V I/O limited.

6	5	4	3	2	1
12	11	10	9	8	7

Figure 1. PMOD connector pinout.

Whatever you connect to them will need to be able to use a 3.3V signal, and send back a 3.3V signal to the board. **Figure 1** shows the pin number arrangement for each of the ports: **JA, JB, JC, JD,** and **JE**.

This probably isn't what you expected the pin numbering to be. This is because the Cerebot series connectors can be either six- or 12-pin. This numbering scheme preserves the pin ordering, whether the connector is six or 12 pins. Pins 5 and 11 are ground and pins 6 and 12 are either 3.3V or 5V, depending on which pins you choose to jumper to select the voltage out. This means that

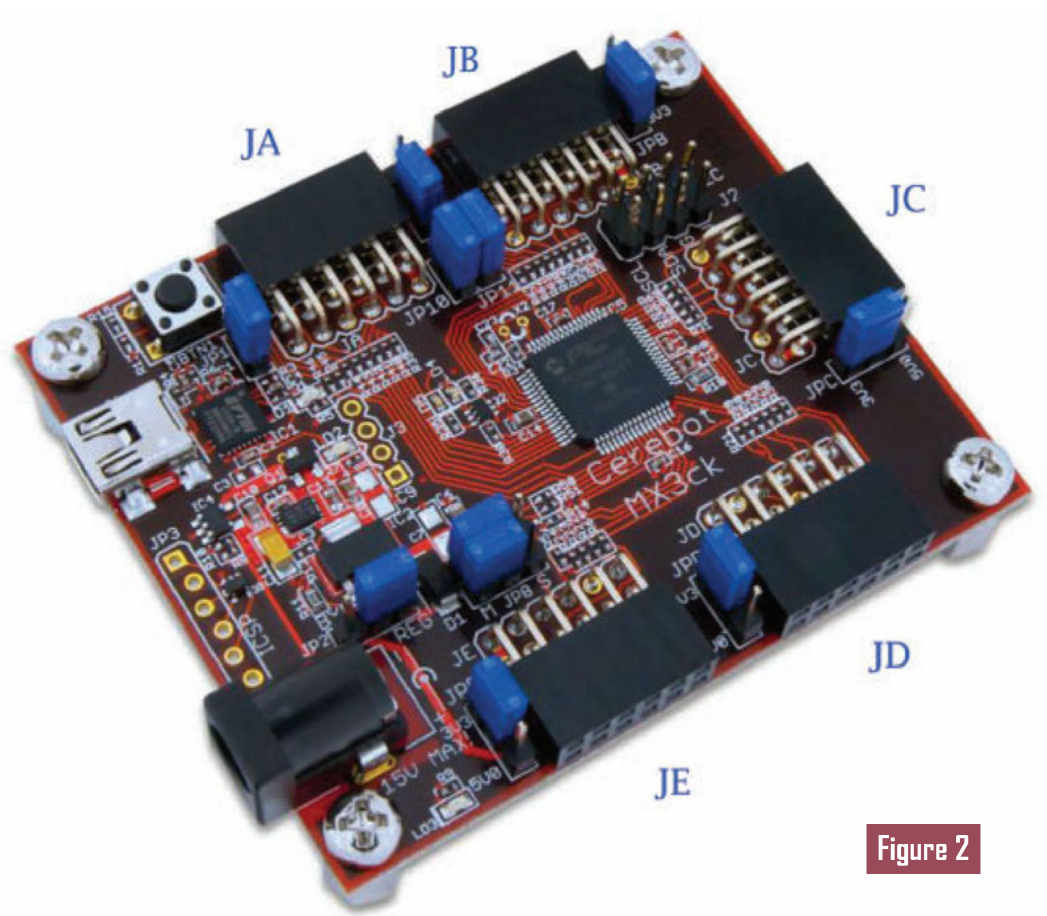


Figure 2

MPIDE I/O	Conn	Pin	PIC ID	Description, special abilities of this port.
0	JA	1	RE0	
1	JA	2	RE1	
2	JA	3	RE2	
3	JA	4	RE3	
4	JA	7	RE4	
5	JA	8	RE5	
6	JA	9	RE6	
7	JA	10	RE7	
8	JB	1	RD9	
9	JB	2	RF3	UART1TX
10	JB	3	RF2	UART1RX
11	JB	4	RF6	
12	JB	7	RD6	
13	JB	8	RD5	
14	JB	9	RD4	PWM
15	JB	10	RD7	
16	JC	1	RB8	A0
17	JC	2	RF5	UART2TX
18	JC	3	RF4	UART2RX
19	JC	4	RB14	A1
20	JC	7	RB0	A2
21	JC	8	RB1	A3
22	JC	9	RD0	PWM
23	JC	10	RD1	PWM
24	JD	1	RB2	A4
25	JD	2	RD2	PWM
26	JD	3	RD10	
27	JD	4	RB9	A5
28	JD	7	RB12	A6
29	JD	8	RD03	PWM
30	JD	9	RD11	
31	JD	10	RB13	A7
32	JE	1	RG9	SS2
33	JE	8	RG8	SDO2
34	JE	3	RG7	SDI2
35	JE	4	RG6	SCK2
36	JE	7	RD8	
37	JE	8	RB5	A8
38	JE	9	RB4	A9
39	JE	10	RB3	A10
40	J2	3/4	RG3	SDA1 (ON I²C SIX-PIN CONNECTOR)
41	J2	1/2	RG2	SCL1 (ON I²C SIX-PIN CONNECTOR)
42			RF0	PIN_LD1 (LD4)
43			RF1	PIN_LD2 (LD5)

Table 1. UNO32 I/O map to MX3Ck.

every connector has four power and eight I/O pins.

Figure 2 shows where all of the connectors are on the MX3cK. (Sometimes the silkscreen is a bit difficult to read.)

To find our Arduino-like MPIDE I/O lines and associate them with connectors and pins, we need to learn a little about how the MPIDE does its magic. MPIDE abstracts any particular board layout such that anyone can make a board and put their I/O lines wherever it makes the most sense.

There are two files associated with this board definition: *Board_Data.c* holds the mapping tables that define the pinout for a board; *Board_Data.h* defines any special relationships these pins have, like being an LED or a UART or SPI port. These files are located here:

<MPIDE install folder>/hardware/pic32/variants/ Cerebot_MX3cK

Where *<MPIDE install folder>* is depends on your system as to where it will be installed. On my OSX system, for instance, it is in the Applications folder as **MPIDE.app**. You would "right-click" on this application and select *Show Package Contents* and follow this whole path:

Contents->Resources->java->hardware->pic32- >variants->Cerebot_MX3cK

Most of this information can also be found in *Appendix C* of the Cerebot_MX3cK_rm.pdf file found at **www.Digilentinc.com**. I guess you could say that this is a "teachable moment;" the manual is your friend!

Now, let's look at the I/O pins and where they are for now. **Table 1** shows this association.

As you can see, not all of the UNO32 I/O numbers map to an MX3cK connector. There are some signals that don't map to PMOD ports, and two signals that map to LEDs. You can't write a sketch on an UNO32 and expect it to work on an MX3cK.

Whew! That was a bit of an eye-strain to sort out! You can bet I'll be referring to it often if I use this board with the MPIDE.

Well, that's the end of another Mr. Roboto. Hopefully, I've helped and made someone's life a little easier. Next month, I promise I'll have a Roomba running around the room controlled by an Arduino or a chipKIT board! As always, you can contact me at roboto@servomagazine .com and I'll do my best to help you. **SV**

NEW PRODUCTS

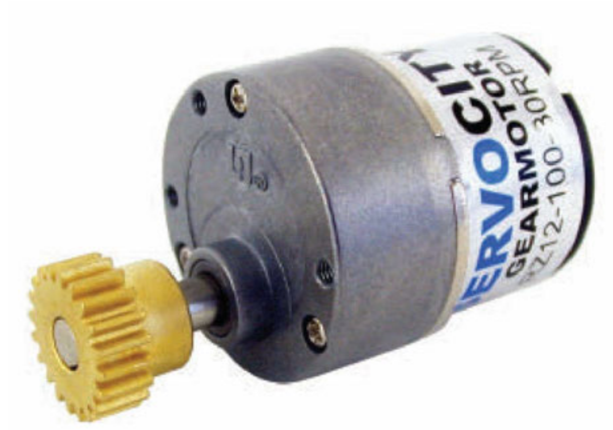
HEX WHEEL ADAPTORS

Adding to their line of ABS and foam wheels, ServoCity has two new hex wheel adaptor sets: one designed for 12 mm hex wheels, and one specifically for 17 mm hex wheels. These wheel adaptor kits allow users to convert the 12 mm or 17 mm hex drive wheels to the ServoCity 0.770" hub pattern that is found throughout their product line.

The recessed 0.770" pattern also allows users to run 6-32 socket head screws into any 0.770 hub or other component that utilizes the same pattern.

Each kit consists of two aluminum mounts, two washers, and two mounting screws. Constructed of 6061-T6 aluminum for strength and durability, these adaptors make it easy to customize mobile projects. Retail price is $9.99/pair.

32P PINION GEARS



ServoCity's selection of gears continues to grow with their new 32 pitch pinion gears that are available in either a 0.250" bore or 6 mm bore. Simply slide the gear on a gearmotor or other compatible shaft and tighten the 10-32 set screw. The gears have a 20 degree pressure angle and are manufactured from hardened brass gear stock. Available sizes include: 16 tooth, 20 tooth, 24 tooth, and 32 tooth. The 32 tooth gear is machined out for weight savings. These pinion gears are ideal for robotics, videography, and R/C applications. Retail price is $12.99/gear.

For further information, please contact:

ServoCity	Website: **www.servocity.com**

CONTACTLESS GESTURE RECOGNITION TECHNOLOGY

A new, low cost contactless gesture-recognition technology called imPart™ which uses Plessey Semiconductor's award-winning EPIC™ electric potential sensors is now available from Saelig. imPart technology is designed to fit around computer monitors, portable laptop computers, ultra books, and tablets to facilitate the control of devices using "no touch" gestures such as swipe, flick, up, and down. A simple 'left-click' function is accomplished by hovering the hand at any required point for more than 0.5 s. Plessey's unique low power EPIC sensors detect changes in electric potential, and allow devices to be gesture-enabled without needing capacitive touchscreen technology or video cameras.

This new technology can detect very small signals even at a distance of up to 4" without physical contact.

EPIC sensors work by detecting changes in the earth's atmospheric electric potential field caused by a hand or

body moving through it. The technology functions as an ultra-high, input impedance sensor that acts as a highly stable, extremely sensitive contactless voltmeter to measure tiny changes in the electric field down to millivolts.

EPIC sensor datasheets are available at **www.plesseysemiconductors.com/products/epic/datasheets/**

For further information, please contact:

Saelig	Website: **www.saelig.com**

AIR MODULE FAMILY FOR ZIGBEE AND BOOSTER PACK KIT

Anaren, Inc., has introduced a family of four Anaren Integrated Radio (AIR™) modules designed specifically to help OEMs develop products that wirelessly communicate in compliance with the ZigBee® standard.

Based on the Texas Instruments (TI) CC2530 low power RF SoC (which operates using TI's Z-Stack™ firmware), the new family of AIR modules is bundled with AIR Support for ZigBee — a solution that includes time-

NEXT-GENERATION AUV

Designed for Survey Applications

Hydroid, Inc., has announced its newest AUV: the REMUS 600 Survey (REMUS 600-S). The REMUS 600-S is a high performance version of the successful REMUS 600 AUV and features advanced technology that has only previously been available in Kongsberg's HUGIN vehicles, making it ideal for applications such as International Hydrographic Organization (IHO) quality surveys.

Specifically, the REMUS 600-S incorporates a Kongsberg EM 3002 Multibeam Echosounder (MBES) operating at 300 kHz. The MBES has previously been shown to exceed the feature-detection requirements of the IHO standard S-44 for Order 1a surveys.

The vehicle also features Kongsberg's Navigation Processing Suite (NavP) which improves timing accuracy to 1 ms or better, and provides complete time synchronization of all onboard sensors; and the Navigation Laboratory (NavLab) software which enhances the NavP by increasing navigational integrity.

The REMUS 600-S benefits United Nations Convention on the Law of the Sea (UNCLS) surveys, search and recovery operations, emergency response surveys, rapid environmental assessment, and military survey operations by providing a transportable and reliable platform to gather high resolution data. The vehicle comes equipped with a full suite of communications options, including Iridium, Wi-Fi, and acoustic communication; Hydroid's flexible Vehicle Interface Program (VIP); and an EdgeTech 2200-S Sidescan sonar.

Hydroid's REMUS AUVs are modular and may be fitted with a large number of different types of sensors, and have been used to aid in hydrographic surveys, harbor security operations, scientific sampling, and mapping, as well as many basic and applied research programs funded by ONR, DARPA, and the UK Ministry of Defense.

For further information, please contact:

Hydroid, Inc.	Website: **www.hydroid.com**

saving AIR-ZNP firmware (including 30+ code examples), pre-certification to applicable global, regulatory standards, and development tools like the company's new BoosterPack for TI MSP430™ and Stellaris® LaunchPad development kits.

"Not everyone seeking to comply with the ZigBee standard is prepared for the programming and RF complexities involved with this exciting wireless protocol," says Anaren Business Development Manager, Mark Bowyer. "With our new AIR modules — and with the many value-added elements of AIR Support for ZigBee solutions that surround these modules — Anaren offers OEMs heading down the ZigBee path a clearer, more streamlined, and more cost-effective approach."

Features and benefits of Anaren's new AIR module family for ZigBee standard applications (part number A2530x24xxx) include the following:

General:
- Minimal RF engineering and ZigBee experience necessary
- Easy to program, for shortened design cycles
- Choice of integral or connectorized antenna
- Pre-certified to FCC/IC, compliant with ETSI
- Choice of range extender or non-range extender modules
- Tiny standardized footprint: 11 x 19 x 2.5 mm
- 2.4 GHz IEEE 802.15.4 compliant RF transceiver
- Excellent receiver sensitivity and robustness to interference (-95 dBm average)
- Wide input voltage range (2.2V-3.6V)
- 100% RF tested in production
- Module weight approximately 0.7 grams
- Low current consumption
- Three low power/sleep modes from 1 µA to 200 µA

Microcontroller:
- High performance and low power 8051 MCU core with code prefetch; 256 KB in-system programmable Flash; and 8 KB RAM with retention in all power modes.

Firmware:
- Preloaded with Anaren's AIR-ZNP firmware, based on the TI Z-Stack for the ZigBee standard (developed in cooperation with Tesla Controls)
- Supports SPI and UART communication
- Driver library included for MSP430 and Stellaris MCUs, which abstracts functionality
- Over 30 code examples for a paired MCU (included to demonstrate functionality)

In concert with the launch of its new AIR module family for ZigBee standard applications, Anaren has also introduced a new BoosterPack featuring its new family of modules. The new CC2530 BoosterPack Kit helps OEM engineers develop wireless applications using a TI LaunchPad for MSP430 or Stellaris MCUs. Key attributes of this new BoosterPack include the following:

Benefits:
- Provides "out of the box" wireless connectivity for easier development of applications based on the ZigBee standard.
- Includes the AIR-ZNP firmware solution (based on TI's Z-Stack) which compresses time to market by greatly reducing the learning curve and development time.
- Provides a learning/development tool for all levels of ZigBee expertise, as well as a clear, easy-to-understand migration path from development to production.

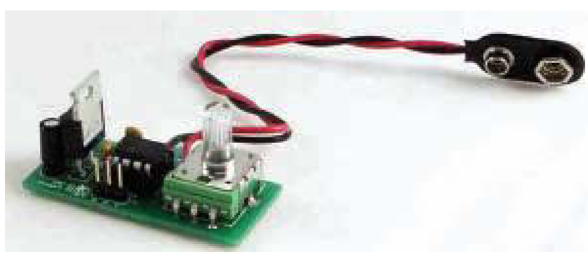
Kit Contents:
- Three A2530E24A AIR Module BoosterPacks for connection to TI's MSP430 or Stellaris LaunchPad development kit (LaunchPad not included).
- An on-board MSP430G2553IN20 Value Line MCU, pre-flashed with Anaren's AIR-ZNP firmware (based on TI's Z-Stack for the ZigBee standard).
- For Stellaris operation, simply remove the MSP430 MCUs, and load the Stellaris firmware via USB from the included CD.
- CD contains all the software, MSP430 and Stellaris drivers, application notes, quick start guide, and more to get started.
- One 2xAA battery holder for remote.

For further information, please contact:

Anaren	Website: **www.anaren.com**

SERVO MOTOR CONTROLLER

Images SI, Inc., has released a updated servo motor controller that provides smoother servo motor response. The SMC-01 is a manual controller for a single servo motor by way of an onboard potentiometer. The heart of the SMC-01 is a PIC12F683 microcontroller. The potentiometer connects to the microcntroller and proportionally controls the servo motor's rotation.



The servo motor shaft will respond as fast and as far as the potentiometer knob is turned. A universal three-pin header makes it easy to connect servo motors; just plug them into the board. The circuit is controlled by the inexpensive eight-pin microcontroller and is powered by a nine volt battery. Servo motors and batteries are not included.

This unit can be purchased as a kit or fully assembled. The list price for the SMC-01A (assembled unit) is $34.95; the SMC-01 (kit) is $24.95.

For further information, please contact:

Images, Co. Website: **www.imagesco.com**

HUMAN INTERFACE BOARD

The Human Interface Board from Parallax brings advanced functionality to the QuickStart development platform. The Human Interface Board stacks directly on top of the QuickStart development platform, creating a minimal footprint for both boards.

With multiple audio and video outputs, and keyboard or mouse inputs, as well as a microSD card socket, the Interface Board can add an interactive interface to any project. Where applicable, it uses the same I/O pin assignments as the Propeller Demo Board peripherals.

Like the QuickStart development platform, the Human Interface Board is an open-source hardware design, so all design files — including layout, schematics, and firmware — are available under licenses that allow free distribution and reuse. The Human Interface Board's design can be incorporated into new applications royalty free and without a non-disclosure agreement. Key features include:

- Multiple audio and video outputs for application flexibility.
- Two PS/2 ports for either two keyboards, two mice, or a mix with one keyboard and one mouse.
- Infrared transceiver for remote control and communications.
- Expandable storage for up to 32 GB of data storage and retrieval.

A couple application ideas would be a control system kiosk or an embedded video game system. The retail price is $29.99.

For further information, please contact:

Parallax Website: **www.parallax.com**

Solar Space Fleet Kit

The Solar Space Fleet Kit from OWI, Inc., is an innovative solar powered science kit that can transform into seven different lunar modules and can be energized via direct sunlight or micro-rechargeable battery.

The Solar Space Fleet allows builders to snap together parts (no tools required) to create different working models including a space station, robot, rover, vehicle, astronaut, shuttle, and dog. Each complete model is powered by a mini solar panel using sunlight, indoor halogen light, or micro-rechargeable battery. Once built, the different models move around, and speed up or slow down depending on the intensity of the light. The kit can be powered by either a solar panel or a micro-rechargeable battery (both the solar panel and micro-rechargeable battery are included in the kit). The micro-rechargeable battery can be charged in two ways: direct sunlight or alkaline batteries (AAA x 2, not included). The suggested retail price is $28.95 USD.

For further information, please contact:

Owi Robots Website: **www.owirobots.com**

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**.

— R. Steven Rainwater

FEBRUARY

1-3 **Quark Roboficial**
Goa, India
Student competition for autonomous robots with a variety of events.
www.bits-quark.org

1-4 **Robotix**
West Bengal, India
Events at the competition include Abyss, Overhaul, A.C.R.O.S.S., Lumos, The Seeker, and Marauder's Map.
www.robotix.in

15-17 **Motorama Robot Conflict**
Harrisburg, PA
R/C vehicle operators destroy their expensive toys for your amusement.
www.nerc.us

28 **Pragyan**
National Institute of Technology, Trichy, India
Student competition. Events include Traffic Rush, Fix the Android, Diner Dash, and Lanka Reloaded.
www.pragyan.org

MARCH

8-9 **AMD Jerry Sanders Creative Design Contest**
University of Illinois at Urbana-Champaign, IL
Autonomous robots built by college teams compete in one of the oldest university level robot competitions.
http://jsdc.ec.illinois.edu

8-10 **Cognizance**
IIT Roorkee, India
A variety of events for autonomous robots.
http://cognizance.org.in

9-10 **METU Robotics Days**
METU Culture and Convention Center, Turkey
Robots compete in events that include line-following, Sumo, mini Sumo, multi-mini Sumo, minefield, triathlon, and a new search and rescue contest.
http://topluluk.odturobotgunleri.org.tr

14-16 **Festival de Robotique**
Stade Uniprix, Montreal, Quebec, Canada
FIRST events include FRC, FLL, and jrFLL.
http://festivalderobotique.ca

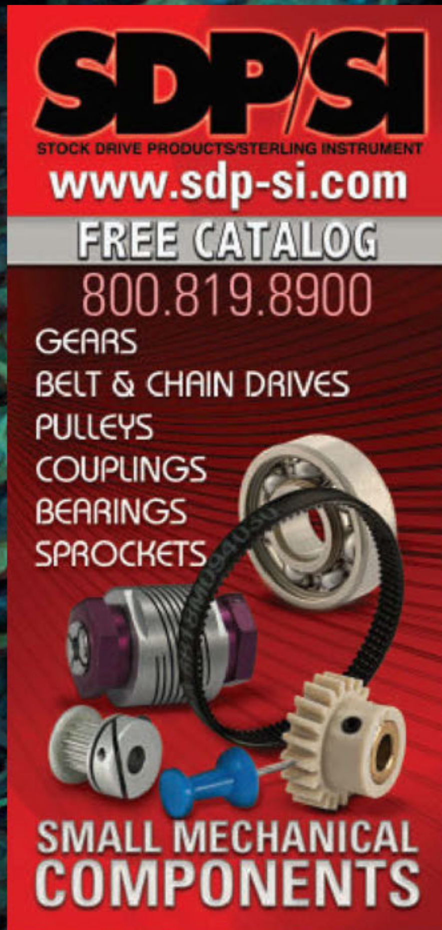
14-17 **Techkriti RoboGames**
Indian Institute of Technology, Kanpur, Uttar Pradesh, India
Multiple events for autonomous robots.
www.techkriti.org

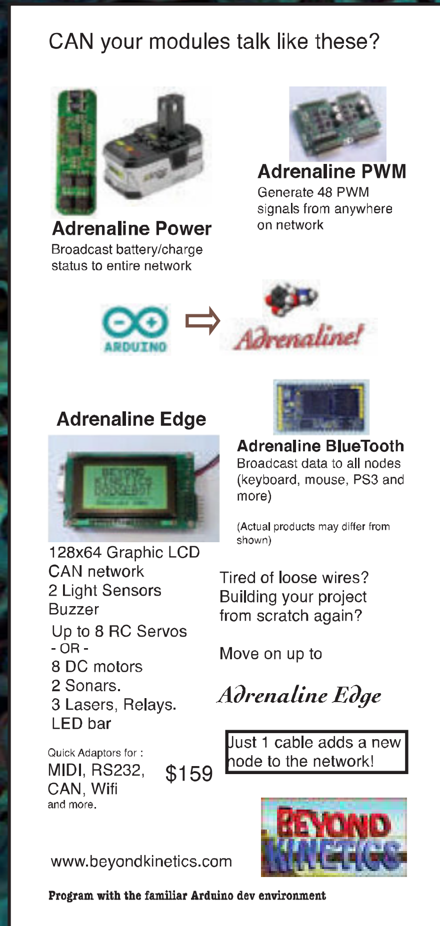
15-19 **Apogee iStrike**
Zuarinagar, Goa, India
Check the website for forthcoming details on this year's robot events.
www.bits-apogee.org

16 **Manitoba Robot Games**
Tec Voc High School, Winnipeg, Manitoba, Canada
Events include Sumo, Mini Sumo, Mini Tractor pull, Super Scramble, Line Following, and Robo-Critters.
www.scmb.mb.ca

17-21 **APEC Micromouse Contest**
Long Beach, CA
Super-fast 25 cm square micromouse robots compete in maze solving.
www.apec-conf.org

23-24 **RobotChallenge**
Vienna, Austria
Lots of events for autonomous robots including standard Sumo, mini Sumo, micro Sumo, the Parallel Slalom, and the Slalom Enhanced.
www.robotchallenge.org





Robotics Showcase

bots IN BRIEF

WATER WANDERER

iRobot is now offering a brand new pool cleaning robot: the Mirra 530.

The Mirra 530 is actually the third model that they've come out with, after the Verro 300 and the Verro 500. At $1,300, it's also the most expensive. If you can afford it, it looks like this robot will probably keep your pool squeaky clean without you having to lift much more than a finger or two.

Features and benefits include:

• Dual top-load filter canisters for easier and faster emptying of debris.
• Modern styling.
• iAdapt Nautiq allows the robot to size up the approximate dimensions of the pool, then choose the optimum cleaning cycle for maximum efficiency.
• Provides thorough cleaning by making multiple passes over the entire inside of the pool, floor to waterline, including walls and stairs.
• Constantly responds to its environment by navigating obstacles, changing directions when necessary, and making sure not to tangle its 60 foot floating power cord.
• Self-contained vacuum, pump, and filter system works without suction lines or other modifications to existing systems.
• Pumps and filters 70 gallons of pool water per minute.
• Reduces heated hot spots and keeps chlorinated water evenly distributed.
• PVC active scrubbing brush lifts dirt, algae, and bacteria off the pool floor and walls, including the water line where oils and bacteria can build up. Fine filters capture debris as tiny as two microns.

SWEET THOUGHTS

Whatever one may think of the military's pronounced interest in robots, it does end up paying for all kinds of things, and here is one amazing example that's funded by DARPA (among others): a seven DoF brain-controlled robotic arm and hand that has allowed this woman to take a bite of chocolate unassisted for the first time in a decade.

Jan Scheuermann has had quadriplegia for the last 10 years, which means that she can move her head and neck, but nothing lower down. She's part of a program at the University of Pittsburgh School of Medicine which involved implanting electrodes directly into her brain, and then using them to intercept brain signals that Jan can use them to control a robotic arm.

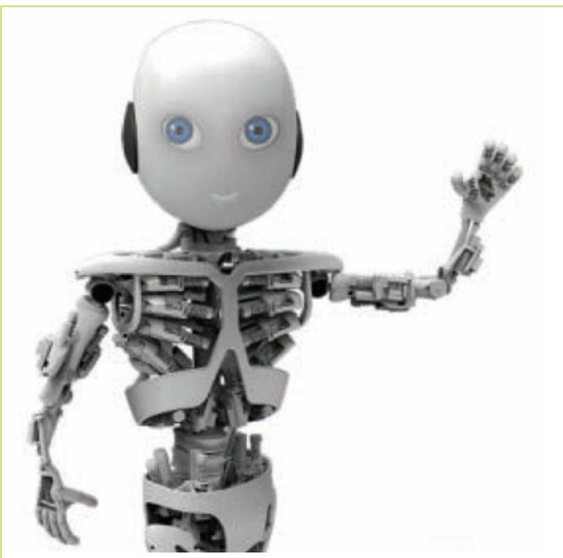
"I used to have to think, 'up, clockwise, down, forward, back ...' Now, I just look at the target, and Hector [the arm is called Hector] goes there."

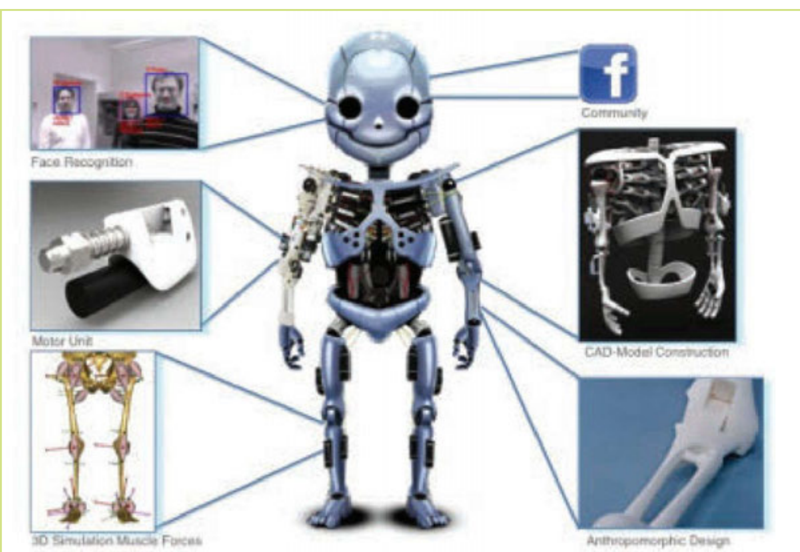
bots IN BRIEF

CLEANER SWEEP

Samsung's new vacuum — the Smart Tango Corner Clean — has pop-out brushes that are designed to help the round robot get dirt and dust out of square corners.

This whole "round vacuums in square rooms" issue is why the Neato has a square front. Roombas try to solve the problem with a spinny brush, but since the brush is mounted on the robot's round chassis, its reach is limited. The brushes pop out whenever the Tango senses that it's in a corner, allowing it to more effectively get dirt out of places where it doesn't really fit.





I AM A REAL BOY

Engineers from the University of Zurich's Artificial Intelligence Lab have begun a rush job on a service robot designed to look like a humanoid boy that will be unveiled March 2013. Roboy — a 1.2 m tall bot — will have artificial tendons for smoother actions, and soft skin. His face will be designed after taking a poll to help decide what it should look like. While the project has 15 partners and 40 engineers, the creators are hoping that it can find other backers.

A donation of $55,000 means adoption and your logo on Pinocchiobot.

Check out a photo gallary at **www.gizmag.com/roboy/25571/pictures**.

SUB CHASER

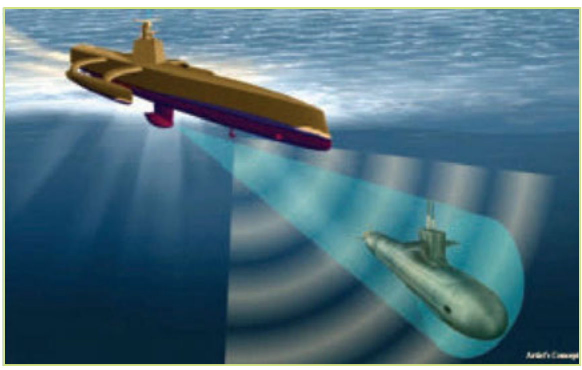
The Science Applications International Corporation (SAIC) has received a $58 million contract with DARPA to construct an Anti-Submarine Warfare Continuous Trail Unmanned Vehicle to chase down evil submarines. Promising a 60 to 90 day stint out in the water, the ACTUV will track with sonar and other sensors, and link back to the Navy ship via satellite. It will then follow the sub to its port or return to its own.

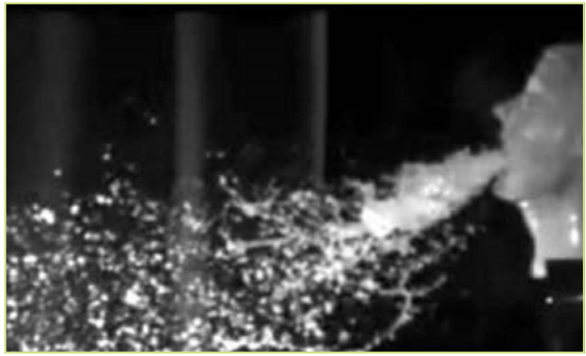
It's important to note that the contract given is for a technology demonstration — not a program of record. The ACTUV will help the Navy mature technologies useful for future capabilities but is not expected to enter active fleet service itself. According to DARPA's ACTUV website, the first completed phase refined and validated the system concept and associated performance metrics, completing risk reduction testing to inform program risks associated with submarine tracking sensors and maritime autonomy.

SAIC is tasked with phases 2-4 — specifically to design a vessel (phase 2); build a vessel (phase 3); and test the vessel (phase 4). Operational prototype at-sea testing is expected in mid-2015.

As stated in DARPA's press release, the goal of the program is an "unmanned vessel that tracks quiet diesel electric submarines for months at a time, spanning thousands of kilometers of ocean with minimal human input." The website adds that an objective of generating a vessel design that "exceeds state-of-the art platform performance to provide complete propulsive overmatch against diesel electric submarines at a fraction of their size and cost." In other words, the vessel must be small and cheap, yet robust enough to operate for 80 days and 6,200 km without human maintainance or refueling.

The approach the program takes for propulsion will be interesting to see develop, as most long-range drone concepts have relied on solar panels or wave propulsion at the sacrifice of top speeds. Part of ACTUV's endurance and speed will come from the drone's design. According to **navaldrones.com**, the SAIC-built concept uses a trimaran hull for better speeds over long ranges as opposed to traditional monohull designs. Additionally, going sans-crew frees up space (normally devoted to crew-support systems) to fill with more fuel tanks.





BRACE FOR BOT BARF

Until recently, the grossest robot ever was Ecobot (which poops). This robot is much, much grosser. Its name is Vomiting Larry, and it's designed to do one thing: puke just like a human.

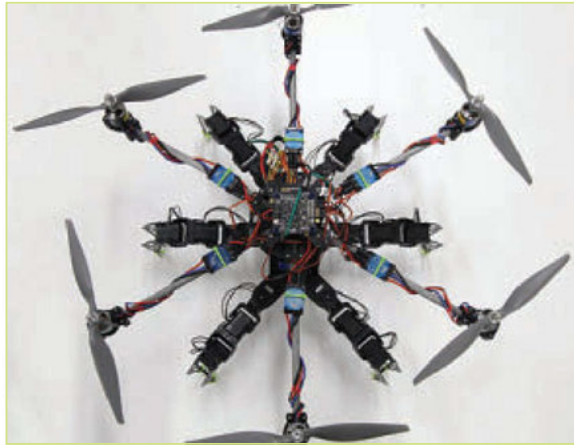
Vomiting Larry is a humanoid simulated vomiting system. He may be the only humanoid simulated vomiting system in existence, but we certainly don't need more than one. Vomiting Larry is doing some important work, though. He's being used to research the spread of noroviruses, which cause humans to projectile vomit, spreading the virus all over the place. Here's a description from Wikipedia:

"Vomiting, in particular, transmits infection effectively. In one incident, a person who vomited spread infection right across a restaurant; 126 people were dining at six tables where one woman vomited. Staff quickly cleaned up, and people continued eating. Three days later, others started falling ill; 52 people reported a range of symptoms. More than 70% of the diners on an adjacent table fell ill; at a table on the other side of the restaurant, the rate was still 25%."

Noroviruses can be aerosolized in vomit, and all it takes is a handful of virus cells to infect you. Vomiting Larry's job is to puke its (lack of) guts out, and then researchers get to measure how far the virus travels and at what concentrations over a variety of surfaces to better understand how it's transmitted. (Ewh.)

Noroviruses are responsible for 21 million illnesses in the US every year — second only to the common cold. If you get one, it probably won't kill you, but you can look forward to nausea, watery diarrhea, abdominal pain, loss of taste, general lethargy, weakness, muscle aches, headache, coughs, and a low-grade fever. Oh, and of course, "forceful vomiting."

GET A GRASP(ER)



Different aerial grasping robots from DARPA, UPenn, and Yale are just plain cool. Now researchers from the University of Twente in the Netherlands, and a scruffy-looking hobbyist from Trossen Robotics have developed their own.

The University of Twente is part of the AIRobots project, the goal of which is "to develop a new generation of aerial service robots capable to support human beings in all those activities which require the ability to interact actively and safely with environments not constrained on ground but, indeed, freely in air." (Whew!)

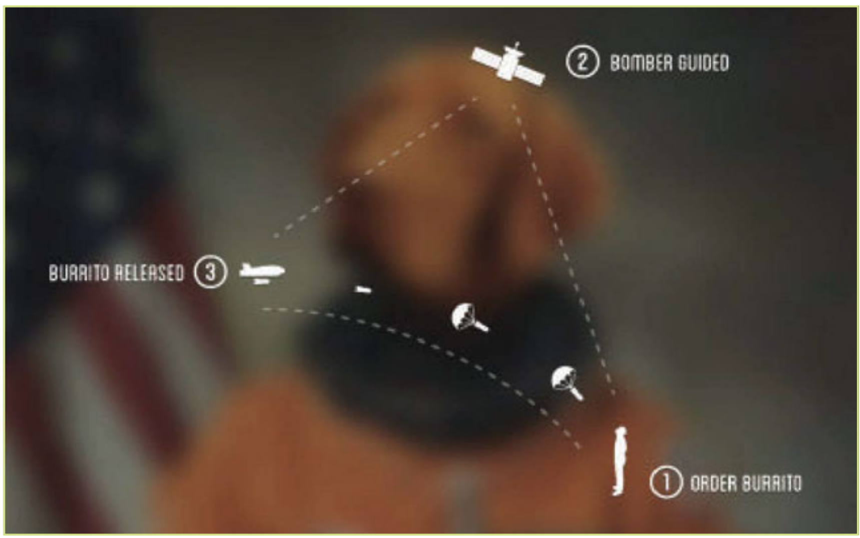
This particular AIRobot is being developed by professor Stefano Stramigiol, Raffaella Carloni, postdoc Matteo Fumagalli, and Ph.D. student Abeje Y. Mersha.

It's kind of amazing how, just in the last few years, plummeting hardware costs and skyrocketing capabilities have enabled that group of geniuses we like to call "hobbyists" to more or less keep up with just about whatever the latest research is — at least when it comes to the hardware itself.

Andrew Alter, one of the instigators of Mech Warfare and current senior executive robot geek at Trossen, writes:

"I designed a modified PhantomX hexapod and we built it out of carbon fiber so it would be light enough to fly. Some friends at Mad Lab Industries are quadcopter gurus, so building the rest of the custom hexacopter was a breeze."

Developing robots that can fly and move along the ground has been a priority for the military for a while, because flying robots are versatile but suck down batteries like nobody's business. While perhaps not the most efficient of compromises, the hexapodocopter is pretty sweet, and there has to be some potential there just for that reason.



HOLY FLYING FRIJOLES!

A team at Darwin Aerospace (DA) must have been too busy to stop working when they came up with the idea of the Burrito Bomber. The idea here is to order a burrito via an app while your phone sends your location. A burrito is loaded into a "delivery tube" which is flown and dropped to you.

It works like this:

You connect to the Burrito Bomber web-app and order a burrito. Your smartphone sends your current location to the DA server, which generates a waypoint file compatible with the drone's autopilot.

The waypoint file is uploaded to the drone and your burrito is loaded into a custom-made burrito delivery tube.

The drone flies to your location and releases the burrito delivery tube. The burrito parachutes down to you, the drone flies itself home, and you enjoy your carne asada.

The airframe is a SkyWalker X8 Flying Wing. The plane uses Ardupilot to navigate the skies. The burrito release mechanism is the combination of a Quantum RTR Bomb System, a 3" diameter mailing tube, and some 3D printed parts that were designed in-house. The plane is controllable either manually via live video transmitted from the plane, or autonomously using the Ardupilot autopilot. A Futaba 9C controller and EzUHF transmitter are used to manually control the plane.

The web app is built in Flask. It gets the user's location via the HTML5 Geolocation API, generates a Mission Planner compatible waypoint file, then sends that to the drone operator. The drone operator uploads the waypoint file to the plane.

Unfortunately, Burrito Bomber is not yet allowed as a commercial product under current FAA guidelines. However, the FAA Modernization and Reform Act of 2012 requires the FAA to hammer out regulations for commercial use drones by September 2015.

MINI MILLI BUG BOT


Photo courtesy of MIT Center for Bits and Atoms

The device in the photo doesn't look like much: a caterpillar-sized assembly of metal rings and strips resembling something you might find buried in a home-workshop drawer. However, the technology behind it and the long-range possibilities it represents are quite remarkable.

This little device is called a milli-motein — a name melding its millimeter-sized components and a motorized design inspired by proteins which naturally fold themselves into incredibly complex shapes. This minuscule robot may be a harbinger of future devices that could fold themselves up into almost any shape imaginable.

The device was conceived by Neil Gershenfeld, head of MIT's Center for Bits and Atoms, visiting scientist Ara Knaian, and postdoctoral associate Kenneth Cheung. Its key feature, according to Gershenfeld is: "It's effectively a one-dimensional robot that can be made in a continuous strip without conventionally moving parts, and then folded into arbitrary shapes."

To build the world's smallest chain robot, the team had to invent an entirely new kind of motor — not only small and strong, but also able to hold its position firmly even with the power switched off. The researchers met these needs with a new system called an electropermanent motor.

The motor is similar in principle to the giant electromagnets used in scrapyards to lift cars, in which a powerful permanent magnet (one that, like an ordinary bar magnet, requires no power) is paired with a weaker magnet (one whose magnetic field direction can be flipped by an electric current in a coil). The two magnets are designed so that their fields either add or cancel, depending on which way the switchable field points. Thus, the force of the powerful magnet can be turned off at will — such as to release a suspended car — without having to power an enormous electromagnet the whole time.

In this new miniature version, a series of permanent magnets paired with electromagnets are arranged in a circle; they drive a steel ring that's situated around them. The key innovation, Knaian explains, is that "they do not take power in either the on or the off state, but only use power in the changing state," using minimal energy overall.

THE DRONERGAMES

DroneGames, which took place recently in San Francisco, CA, tasked programmers with hacking UAVs in the most interesting and creative ways possible.

DroneGames was held at the Groupon offices, and was sponsored by the likes of Groupon itself, Windows Azure, and NodeCopter, which recently started this helihacking movement. Nine teams took part in the competition which was judged by Chris Anderson (of DIYDrones and now 3DRobotics), Dale Dougherty (founder of MAKE), Andreas Raptopoulos (co-founder of Matternet), and a couple other people. It appears entries were judged mostly on awesomeness, and the results certainly reflect that.

In third place was "TooTall Nate," with a hack that lets you control an AR Drone over a cellular network with a Verizon MiFi card, resulting in unlimited range as long as you've got a decent cell connection.

Second place went to a team of freshmen from Stanford, who figured out a way to control lots of different drones with just one computer.

First place went to James Halliday, who wrote a virus that will infect an AR Drone, and then use that drone to infect any other AR Drones it comes across, "causing them to run amok." Or, if you want to be less evil about it, it's a handy way to automatically deploy software onto a bunch of AR Drones at once.

A crowd favorite seemed to be the project in the photo, from engineers at Groupon. They taught a drone to behave itself on the end of a leash — which is neat — but it's also constantly taking pictures and performing facial recognition.

MUSCLE MANIA

Researchers at the University of Tokyo are taking bio-inspired robots to new heights with Kenshiro — their new human-like musculoskeletal robot. They have added more muscles and more motors to their Kojiro robot from 2010, making Kenshiro's underlying structure the closest to a human's form so far.

Kenshiro mimics the body of the average Japanese 12 year old male, standing at 158 centimeters tall and weighing 50 kilograms. Kenshiro's body mirrors almost all the major muscles in a human, with 160 pulley-like "muscles" — 50 in the legs, 76 in the trunk, 12 in the shoulder, and 22 in the neck. It has the most muscles of any other bio-inspired humanoid out there.

Why try and mimic the human body? It turns out that getting a robot's weight right is a tricky problem. Yuto Nakanishi, the head of the project, spoke about the weight problems of Kenzoh, Kenshiro's tendon-driven upper-body robot ancestor. Kenzoh was a hearty 45 kg — just for the upper body. Scaling up, they projected that a full-body Kenzoh could weigh as much as 100 kg.

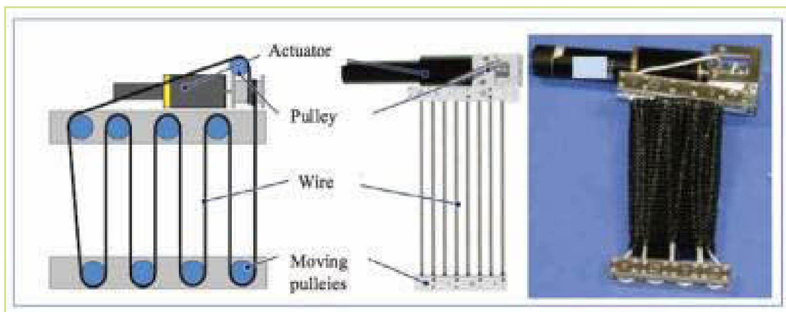
That was a lot of weight for a relatively small robot. So, they decided to design a robot with the same weight ratios of a human. For example, a 55 kg boy would have about a 5 kg thigh and 2.5 kg calf. Kenshiro copies that ratio, with a 4 kg thigh and 2.76 kg calf. Balance is key.

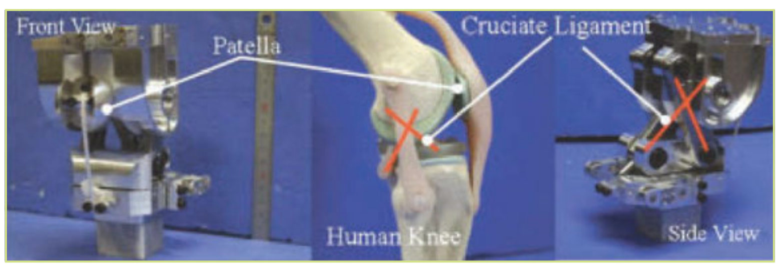
Weight was one thing, but the researchers also tried to mimic the muscle torque and joint speeds. Kenshiro's total power output is five times greater than Kojiro's. Kenshiro can get almost the same amount of joint torque as a human, with joint angular speed not quite at human level — at 70-100 degrees per second. It's a trade-off in weight and power; bigger and stronger motors are often heavier.

Like Kojiro, Kenshiro is actuated by a system of pulley-like muscles. This time, instead of single point-to-point muscles, they decided to make planar muscles.

These flat and wide muscles use only one motor and are much more stable. All in all, these motors give Kenshiro 64 degrees of freedom (except for the hands): 13 in the neck, 13 in each arm, seven in each leg, and 11 in the spine.

Kenshiro's bone structure is also quite amazing. Its aluminum bones — including an impressive rib cage — are sturdier than previous 3D printed bones (breakage tended to be a problem), and its knee-joints include imitations of cruciate ligaments and a floating patella.





Cool tidbits herein provided by www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, http://www.robots-dreams.com, and other places.

ROBOT POWER
www.robotpower.com
Extreme Robot Motor Control
High-Current motor control for Arduino™
Software and plug compatible with our MegaMoto shield
MEGAMOTO Plus Shield
♦ 30A Continuous with fan - 40A peak
♦ Current sensor outputs can be sent to any analog pin
♦ Stack up to three units on one Uno or compatible
♦ Independent half-bridges can control brushless/steppers too!
MADE IN THE USA
Scorpion XXL
♦ 7V - 28V
♦ Dual 40A+ Peak H-Bridges
♦ Current/Temp limiting
♦ R/C inputs w/mixing
♦ Optional enclosure
♦ 3.25" x 2.25" x .5"
Coming Soon!
The Vyper
♦ Single 120A+ H-Bridge
♦ 8V - 42V
♦ R/C inputs w/mixing (2 units req.)
♦ Analog pot input
♦ Supports 8 AWG wires!
♦ Optional enclosure
♦ 2.85" x 2.25" x 1.75"
Phone: 253-843-2504 ♦ sales@robotpower.com

STEER WINNING ROBOTS WITHOUT SERVOS!
Perform proportional speed, direction, and steering with only two Radio/Control channels for vehicles using two separate brush-type electric motors mounted right and left with our mixing RDFR dual speed control. Used in many successful competitive robots. Single joystick operation: up goes straight ahead, down is reverse. Pure right or left twirls vehicle as motors turn opposite directions. In between stick positions completely proportional. Plugs in like a servo to your Futaba, JR, Hitec, or similar radio. Compatible with gyro steering stabilization. Various volt and amp sizes available. The RDFR47E 55V 75A per motor unit pictured above.
www.vantec.com
VANTEC
Order at (800) 882-6832

Model 324
WORLD'S MOST VERSATILE CIRCUIT BOARD HOLDERS
Our line of Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.
MONTHLY CONTEST
Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.
Model 201
PANAVISE®
Innovative Holding Solutions
VISIT US ON
7540 Colbert Drive • Reno • Nevada 89511
1 (800) 759-7535 | www.PanaVise.com

THE ORIGINAL SINCE 1994
PCB-POOL®
Beta LAYOUT
FREE Stencil
with every prototype order
Embedded RFID
authenticate, track & protect your product
NEW!
www.magic-pcb.com
Call Tyler: 1 707 447 7744
sales@pcb-pool.com
PCB-POOL® is a registered trademark of
Beta LAYOUT
create : electronics
www.pcb-pool.com

COMBAT ZONE

Discuss this section in the *SERVO Magazine* forums at http://forum.servomagazine.com

Featured This Month:

BUILD REPORT:

Keres: 150 g Fighting Robot

● by Mike Jeffries

Keres was built to test some prototype Mini-Spark gearboxes from FingerTech Robotics. *(See the accompanying article on page 41.)* After spending some time debating between several weapon systems, I decided that a horizontal spinner with a relatively large spinning bar would be a good test setup.

With a horizontal spinner, you need to be able to move quickly to allow the weapon time to spin up, and to give more options for attack locations. You'll also likely spend a decent amount of time flying around the arena on larger hits, which will be a good way to test how the gearboxes handle sudden impacts without having to intentionally expose your wheels to the other robot.

After settling on the concept, it was time to move to CAD. The specifics of the design changed several times during this period as I saved up to cover the cost of new components, and as part availability changed.

The design uses a laser sintered nylon internal section from Shapeways, and .032" thick titanium top and bottom plates.

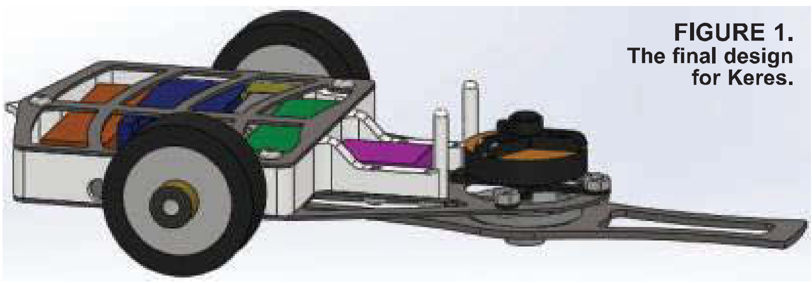


FIGURE 1. The final design for Keres.

www.servomagazine.com/index.php?/magazine/article/february2013_CombatZone

The three pieces are held together using threaded inserts in the main body section and four Plastite screws near the weapon motor.

The electrical system is fairly standard. Keres uses tinyESCs (electronic speed controllers) for drive, a HobbyKing R410 Rx and a two-cell 180 mAh Turnigy Nano-Tech LiPo for power. The weapon motor is a 1560KV GT2203 brushless outrunner from **HiModel.com** and uses a Plush10 ESC from HobbyKing. The weapon motor has been modified via the replacement of the stock shaft with a hardened O1 tool steel shaft.

With the tight weight limit and lack of space necessitated by the design, it took careful planning to fit in all of the components and wire without burning off my fingerprints. The weapon ESC is mounted partially outside of the main body to save space and reduce wire run lengths. Doing this allows for easier access to the motor wires should they ever need to be reversed; it also reduces the overall system weight.

The tinyESCs are mounted between the front wall of the body and the drive motors which also allow for very short wire lengths between the motor and ESC.

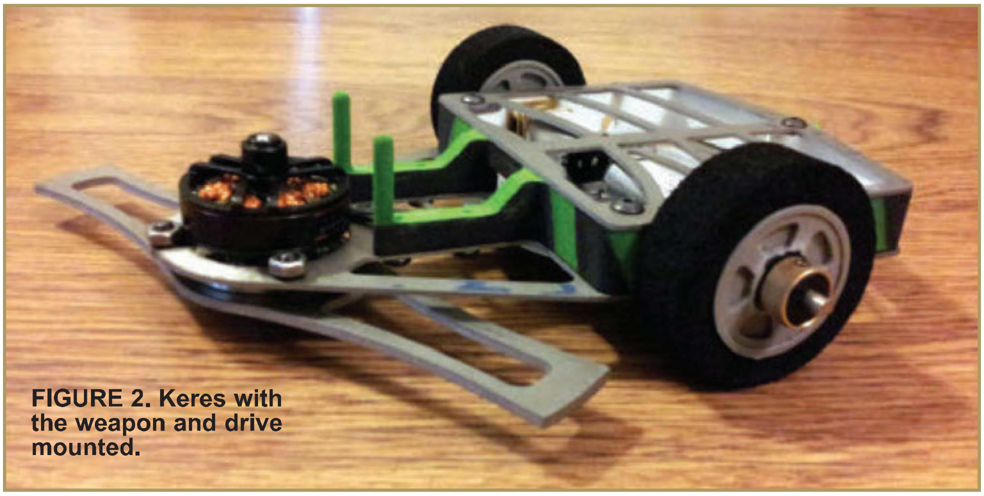


FIGURE 2. Keres with the weapon and drive mounted.

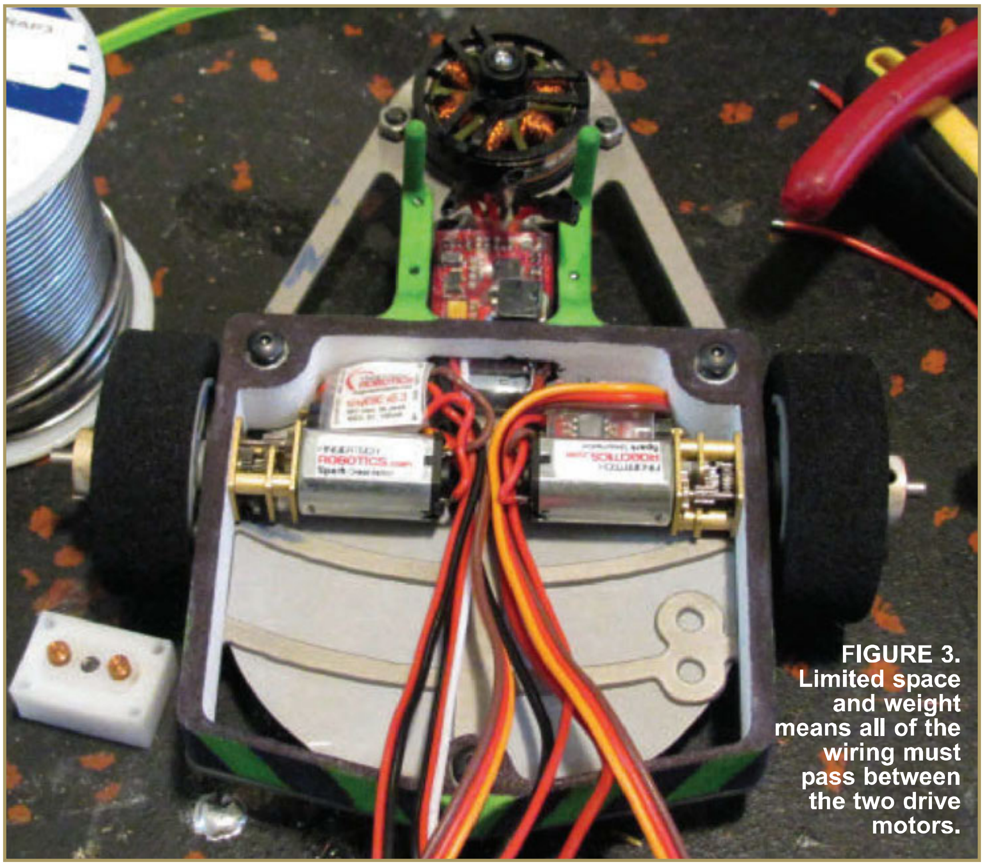


FIGURE 3. Limited space and weight means all of the wiring must pass between the two drive motors.

PARTS LIST

Total cost: $122.19 + materials, labor, and TBA Mini-Spark retail price.

ITEM	QTY	COST	WEBSITE
FingerTech Robotics tinyESC	2	$69.66	**www.fingertechrobotics.com**
FingerTech Robotics Mini-Sparks	2	Prototype	**www.fingertechrobotics.com**
GT2203 outrunner	1	$14.14	**www.himodel.com**
Printed nylon chassis	1	$17.25	**shpws.me/m1b7**
2S 180 mAh Turnigy Nano-Tech battery	1	$4.16	**www.hobbyking.com**
R410 orange Rx	1	$9.99	**www.hobbyking.com**
Lightweight set screw hub 3 mm bore	1	$6.99	**www.servocity.com**
Chassis components .032 titanium	1		**www.nearchaos.net/KeresParts.rar**
Weapon bar .080 titanium	1		**www.nearchaos.net/KeresParts.rar**

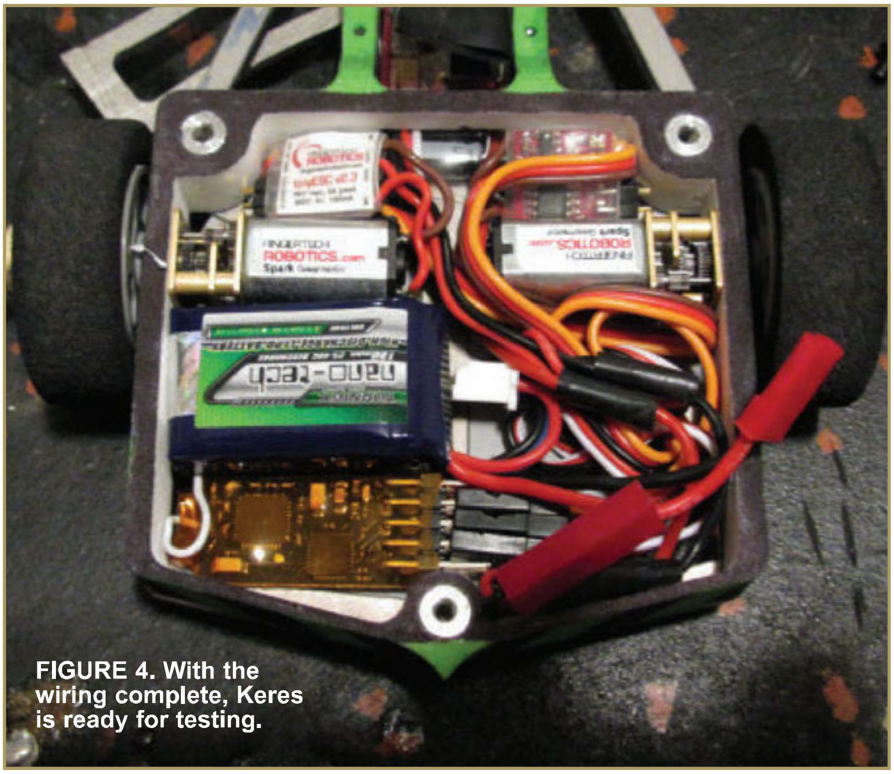


FIGURE 4. With the wiring complete, Keres is ready for testing.

Placing them forward of the motor helps to shift the weight balance slightly forward of the drive wheels, which should reduce how often the front of the robot lifts off of the ground. The receiver and battery are behind the drive motors near a spot that allows for the future addition of a power switch.

An afternoon of soldering later, and Keres was nearly ready for testing. I tossed the batteries on the charger and spent a bit of time "neatening" up the wiring and securing the weapon motor wires in a way that would keep them from getting caught on anything. Not long after that, the batteries were fully charged and Keres was ready for the first test.

Testing went well, and all systems performed as expected. The new drive motors resulted in a very nimble robot, and the weapon appears to pack a decent punch without breaking itself. **SV**

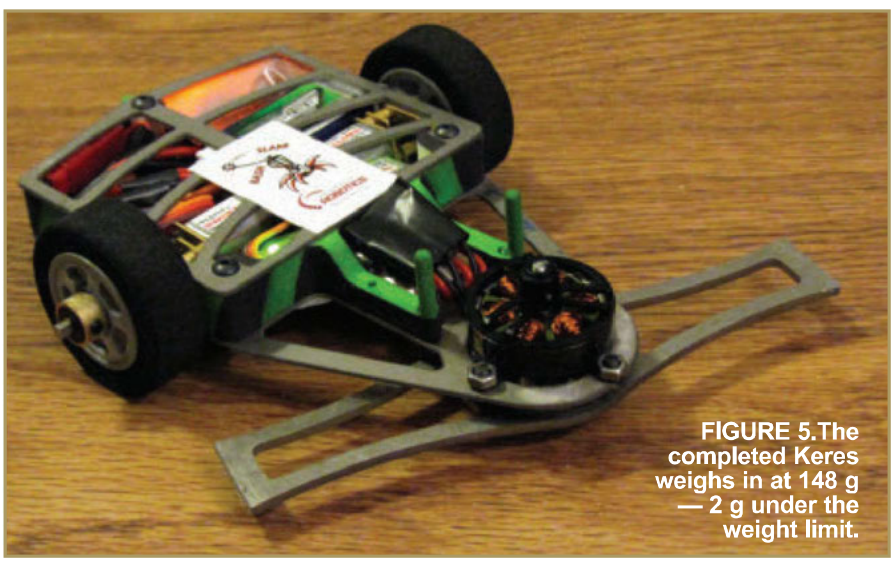


FIGURE 5.The completed Keres weighs in at 148 g — 2 g under the weight limit.

BUILD REPORT:

Mini Bot Hockey Bots: Part 1

● by Pete Smith

Playing bot hockey is great fun, and it's a big hit with the crowds at events. I have my own full team of 15 lb class bots (**Figure 1**), and I detailed their design and build in the October 2011 issue of *SERVO Magazine*.

I also use the bots in the classes I teach each year at summer camp. The kids race and fight Sumo battles with them. It would be great to also have bot hockey matches, but an arena for bots that large is too big for a high school classroom.

The need for something a bit smaller has been in the back of my mind for some time, but it was a request by a couple of customers for a quote for just such a smaller bot that got me inspired.

They wanted something for use at birthday parties, etc., that would fit into a double garage or on a driveway. The bots had to be reliable, easy to repair, and have a long run-time.

I designed a small hockey bot to meet their specifications, but the deal fell through when the full costs of building a fleet of eight+ bots became clear.

However, I decided to go ahead and build a set for use at the summer camps. The kids could build and then compete with the bots (which should keep them busy and happy for at least a couple of days!).

There is an existing "Junior" class of smaller bot hockey bots; details of which can be found at **www.bothockey.com/Bot-Hockey-rules_v0.3.pdf**.

However, I found that it would

FIGURE 1. **Team Scotch Pies.**

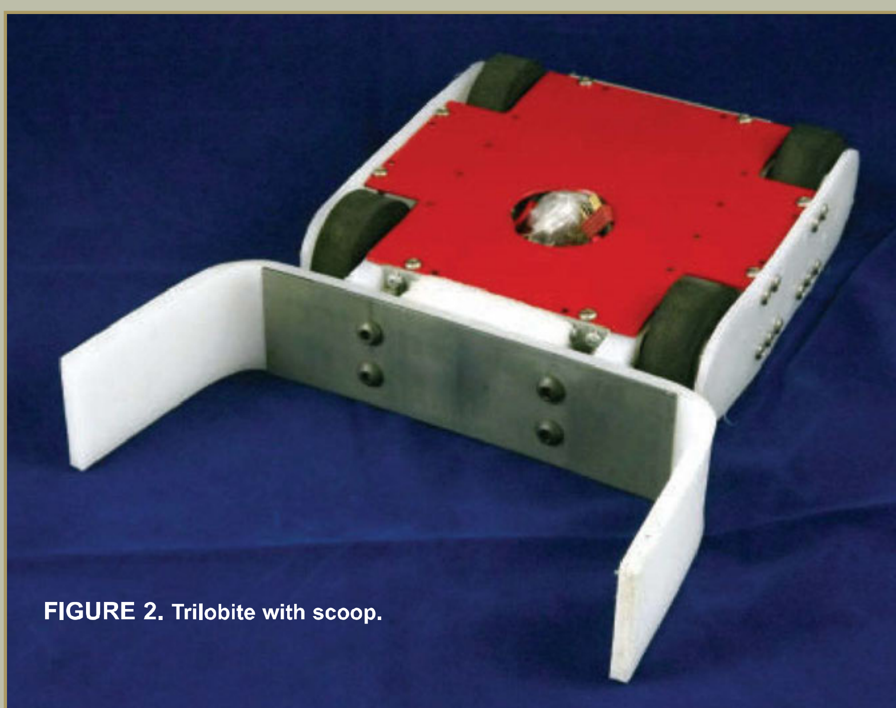


FIGURE 2. **Trilobite with scoop.**

FIGURE 3. LiFe pack.

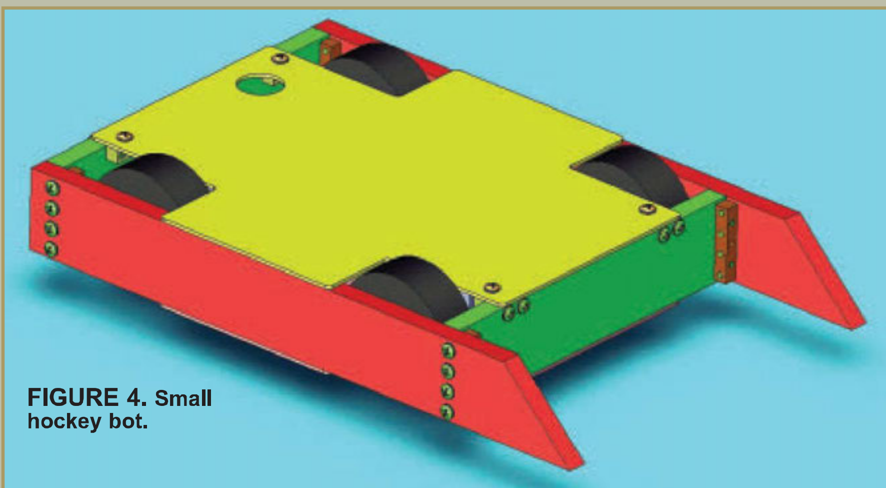


FIGURE 4. Small hockey bot.

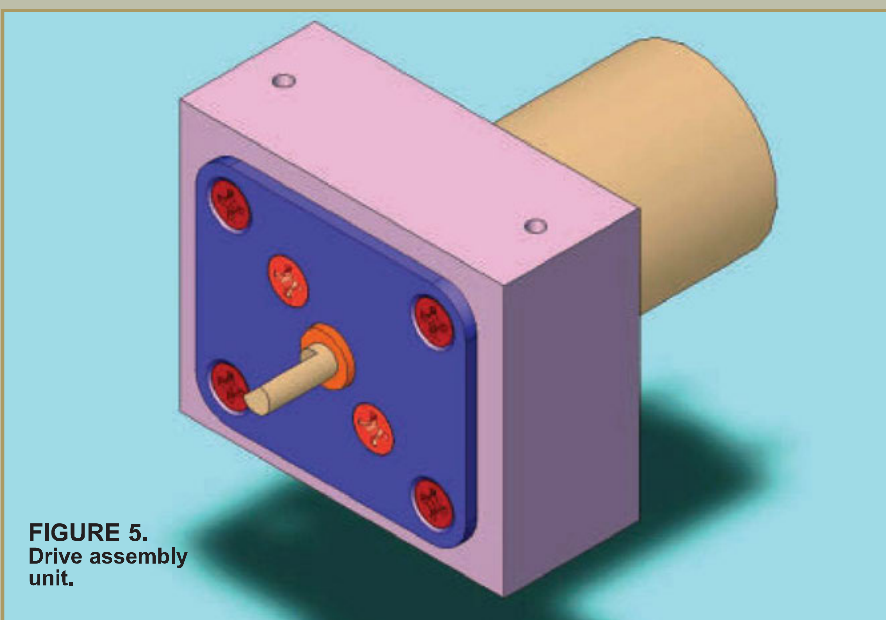


FIGURE 5. Drive assembly unit.

be hard to build in the necessary requirements for a long runtime (i.e., a big battery) in the 7.9" x 7.9" dimensions allowed. As far as I'm aware, no one has actually built to the Junior specifications, and since my bots would only ever be competing against themselves, I decided to build to suit *my* requirements rather than to meet Junior specs.

My first thought was to simply build up a number of my Trilobite kits, and then provide them with simple scoops for helping to catch and guide the puck (**Figure 2**). Trilobite, however, is designed for combat and is much stronger than it needs to be for hockey.

It also uses complex parts and a lot of nutstrip — both of which raise the cost and increase the build time and complexity. Furthermore, it does not have enough space inside for a large LiFe battery pack.

I wanted to use LiFe batteries to reduce the dangers of a battery fire, but they are larger than the equivalent LiPo battery and not available in as many different pack sizes.

I looked at what was available and decided to go for a 3S (9.6V) LFP26650P 2,600 mAh pack, built by Robotic Power Solutions (**www.battlepack.com**). I've used their Battlepacks for years with great success. I ordered a single pack for use in the prototype (**Figure 3**).

The design (**Figure 4**) is based around the motors and electronics used in Trilobite but in a chassis similar to that used in my full sized hockey bots. The main panels would be in 3/8" thick UHMW.

This plastic is tough but easy to work with. It's available in strips of various widths; I chose to use a standard size of 1-1/2 " as the height of each panel. This will greatly reduce the amount of

cutting to make each part.

The top and bottom panels are 1/16" 6061 aluminum. I use a better grade (7075) in my combat bots, but the 6061 should be fine in a hockey bot.

The drive motors are standard Kitbots1,000 rpm Beetle motors with matching mounting plates. I'll mount them in a thicker block of UHMW (**Figure 5**) and secure them to the bottom panel with a couple of screws. This will allow any failed units to be swapped out very quickly.

The chassis is held together by four sections of nutstrip: one in each corner and by the bottom panel (**Figure 6**) which is held in place by 12 self-tapping screws (the other eight screws shown are to secure the drive motors).

The top view (**Figure 7**) shows the four motors in place and also the four short sections of nutstrip which will allow the top to be secured using only four 6-32 screws.

I chose these over self-tapping screws since they will be taken out every time the bot is charged up. Plus, if the nutstrip is damaged, it's easy to replace.

The top panel shown back in **Figure 4** has a 0.9" diameter hole to mount a rocker switch with an integral LED (**Figure 8**) that will be used to power up the bots. I got the switches from All Electronics (**www.allelectronics.com**; catalog number LRS-28B).

Once I was happy with the design, I ordered the required UHMW strip, enough water-cut top and bottom panels for seven bots, and a set of steel templates of the front/back panel, side panels, and the motor mounting blocks.

Part 2 will cover the manufacture of the various parts. **SV**

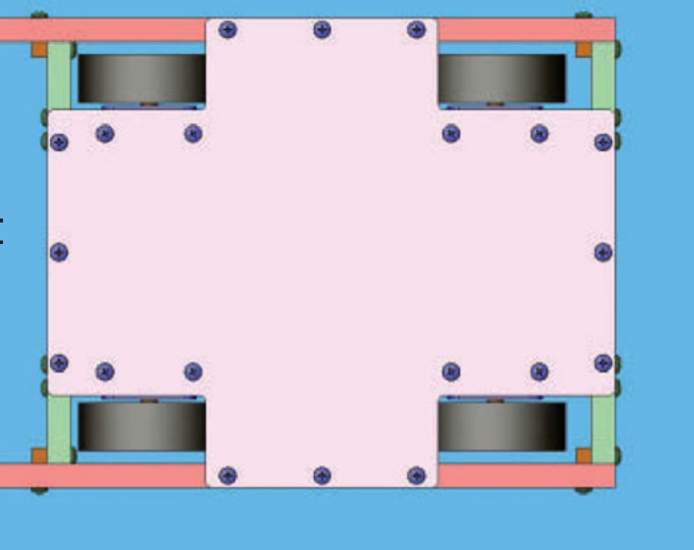


FIGURE 6. Bottom panel.

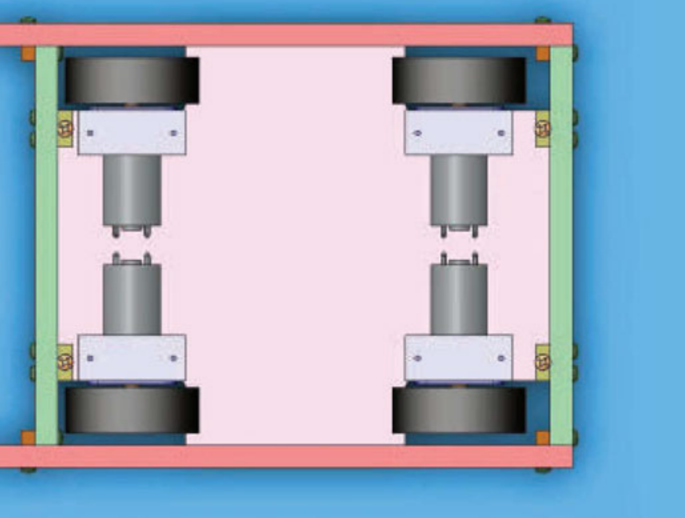


FIGURE 7. Top view.

FIGURE 8. Power switch.

EVENT REPORT:

PennBots — The Second Annual Battle at Yellow Breeches

● by Dave Graham

PennBots — the Robot Club of Pennsylvania — hosted their Second Annual Fall Fling on November 17, 2012 at the Yellow Breeches Middle School in Boiling Springs, PA. The small but dedicated group of fighting robot enthusiasts (**Figure 1**) from Pennsylvania and Maryland met to determine who was the best in three Insect weight classes: Flea (a.k.a., Fairy), 150 grams; Ant, one pound; and Beetle, three pounds. Corporate sponsors *SERVO Magazine*, Pololu, and FingerTech Robotics ensured a bounty of fighting robot mechanical and electrical components awaited the winners.

The Flea competition pitted two big spinners — Hedgehog and Slicer Dicer — against a swarm of wedges. In the end, the wedges would prevail as both spinners

FIGURE 1. PennBots Fall Fling competitors and their bots.

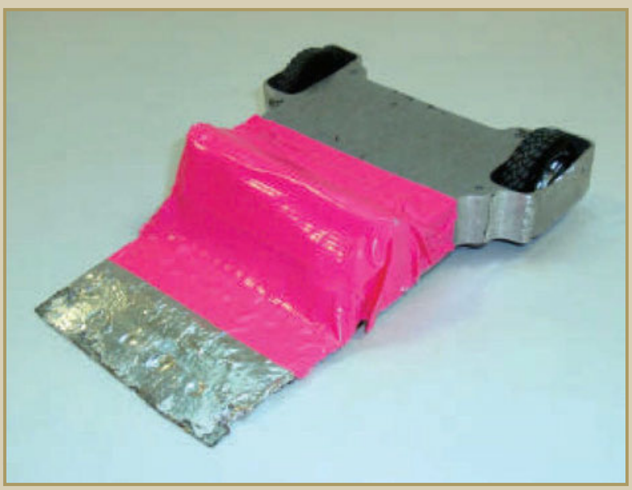


FIGURE 2. Fleaweight bot Baby V.

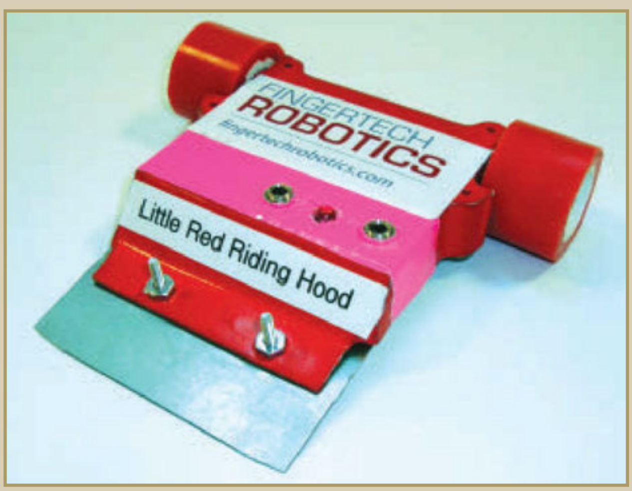


FIGURE 3. Fleaweight bot Little Red Riding Hood.

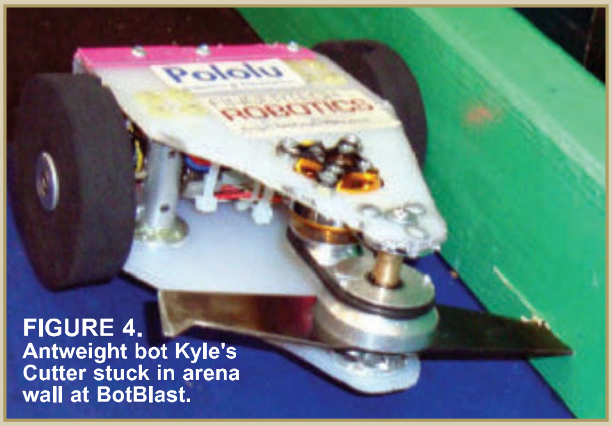


FIGURE 4. Antweight bot Kyle's Cutter stuck in arena wall at BotBlast.

FIGURE 5. Antweight bot Kyle's Cutter stuck in arena floor at the Battle at Yellow Breeches.

experienced mechanical difficulties. Baby V (**Figure 2**) ran the winner's bracket and Little Red Riding Hood (**Figure 3**) defeated Slicer in the elimination bracket. Little Red Riding Hood — powered by FingerTech Robotics tinyESCs and sporting their "arrest me red" Cobra mini Sumo wheels — went on to claim the Fleaweight title.

The Ant competition was exciting and showcased a collection of destructive weapons including a spinning blade, a beater, a mechanical bumper, and the usual group of wedges. My spinning blade Kyle's Cutter ran the winner's bracket but again had problems, sticking the blade into the plywood arena. You may recall this last summer at Jeremy Campbell's BotBlast, Kyle's Cutter stuck in the arena wall (**Figure 4**). The designer of the PennBots arena, Richard Kelley, assured me Kyle's Cutter would not stick in the PennBots arena walls as they were covered in Lexan. However, Kyle's Cutter did go vertical and ended up sticking in the plywood floor (**Figure 5**).

On the way to running the winner's bracket, Kyle's Cutter sent Bump to the elimination bracket and climbed up Cheesus to rip off its power plug and send it south as well (**Figure 6**). In the elimination bracket, Kenney Brandon's big beater Buzz was out early, while his wedge bot Cheesus went on to the final match in the elimination bracket. Newcomer Team Balhogs (**Figure 7**) and their bot Bump with its unique bumper plate met Cheesus in the elimination bracket final.

Bump defeated Chessus to move on to the final match against Kyle's Cutter. Bump took the first match forcing a second fight in the tournament's double elimination format. The second fight was one of the best matches I have ever had.

FIGURE 7. Team Balhogs Colin Foster (left), Will Wooten-Janney (center), and Sam Hanson (right).

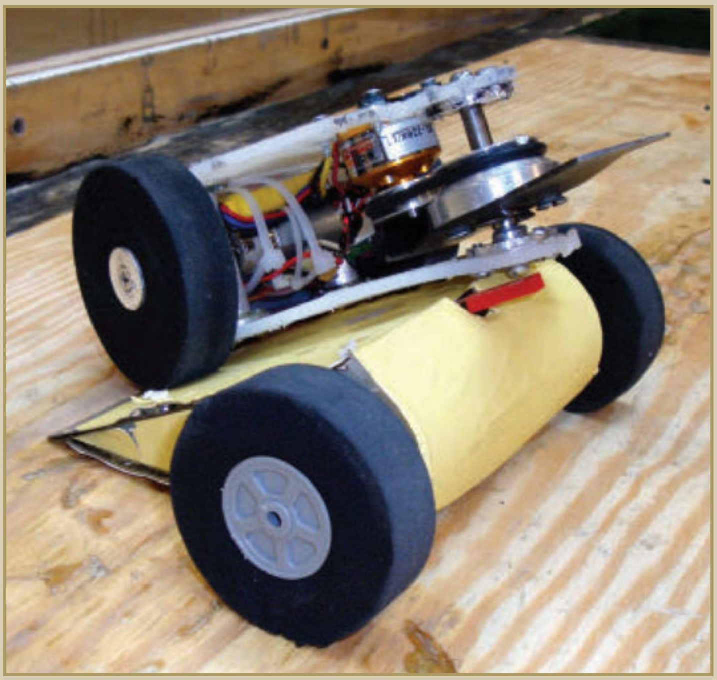


FIGURE 6. Antweight bot Kyle's Cutter climbs up Chessus and rips out its power plug.

FIGURE 8. Antweight bot Bump after fight with Kyle's Cutter.

FIGURE 9. Team Trivial couple Kenney Brandon and Emily Belot.

FIGURE 10. First-time drivers Nadia Slepushkina (left) and Savanna Seabolt (right).

Kyle's Cutter took its best shot and really bludgeoned Bump (**Figure 8**), but Bump hung tough and eventually took the match and the well-deserved Antweight title.

During the Antweight rumble, Team Trivial couple Kenney Brandon and Emily Belot (**Figure 9**) and first time drivers Nadia Slepushkina and Savanna Seabolt (**Figure 10**) did battle. Richard Kelley and his bot Tiger eventually put them all in the pit (**Figure 11**) to win the rumble.

The Beetleweight competition was a round-robin format between wedges. After one round of competition, there was no winner so a second round of competition

FIGURE 11. Antweight bot Tiger put its opponents in the pit during the Antweight rumble.

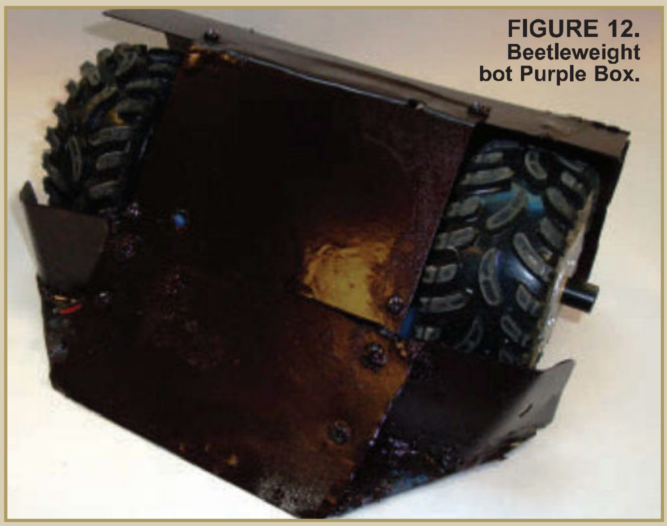


FIGURE 12. Beetleweight bot Purple Box.

TABLE 1 - WINNERS

	FLEA	ANT	BEETLE
1st:	Little Red Riding Hood *Dave Graham*	Bump *Colin Foster*	Purple Box *Richard Kelley*
2nd:	Baby V *Dave Graham*	Kyle's Cutter *Dave Graham*	Sidewinder *Dave Graham*
3rd:	Slicer Dicer *Richard Kelley*	Cheesus *Kenney Brandon*	Box Junior *Dave Kelley*
Melee:	Baby Box *Dave Kelley*	Tiger *Richard Kelley*	Sidewinder *Dave Graham*

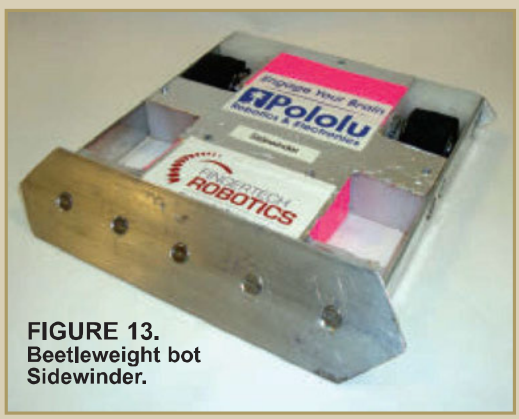


FIGURE 13. Beetleweight bot Sidewinder.

was conducted. Richard Kelley's latest creation, the Purple Box (**Figure 12**), came out on top beating Sidewinder (**Figure 13**) for the Beetleweight title. A complete list of all the winners is in **Table 1**.

Mark your calendars now for the next PennBots event — the Third Annual Downtown Dogfight on Saturday, March 23, 2013 in Harrisburg, PA. This event is a collaborative venture between PennBots and Harrisburg University, and will feature Insect class fighting robots and thinking robot competitions, including line following and maze solving.

This event continues to grow, so register early and plan to attend this great day of robot mania! You can follow PennBots on their website at **www.pennbots.org** and on **www.buildersdb.com**. SV

PARTS IS PARTS:

Product Beta Test: RageBridge from Equals Zero Designs

● by Mike Jeffries

I was lucky enough to get a chance recently to test the new RageBridge dual channel brushed DC motor controller from Equals Zero Designs (**e0designs.com**). I tested the prototype second-generation (approximately v0.2) RageBridge motor controller in my 30 lb robot, Nyx. Beyond not having to build an entire bot for the controller, swapping the new ESC (electronic speed controller) into a working robot means that you're able to directly compare performance between an existing solution and the controller under test.

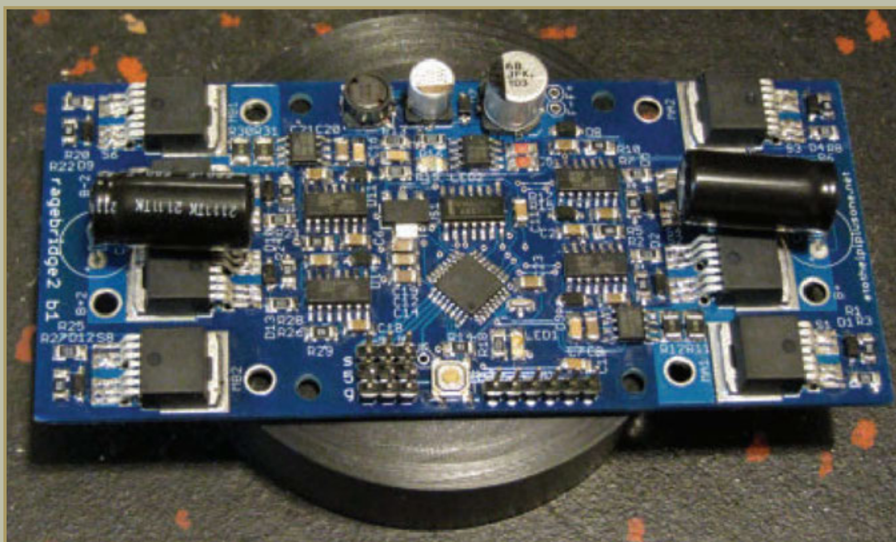


FIGURE 1. RageBridge v2 as delivered for testing.

Before I get into the installation and testing portion, let's start with the specifications:

- Voltage Range: 12V-36V nominal with an absolute 50V limit
- Current Rating: 30A true continuous (40A with fan) and 60A limiting on each side
- Size: 4.5" x 2" x 0.8" and 3.5 oz before adding wires

More detailed specifications and test data can be found at **http://e0designs.com/products/motor-controllers/ragebridge**.

The board arrived ready for testing with only a few elements to handle on my end. The main work to be done was the addition of power and motor leads, and soldering a wire to turn on the onboard mixing for single stick control. (I have been informed that the production RageBridge will use a jumper instead of a soldered wire.) The markings on the board made it easy to figure out which wires needed to attach where, and after a short time I was ready to begin the initial testing.

After hooking up my PWM (pulse width modulation) cables to the appropriate connectors, I swapped the battery and motor leads to the RageBridge from my old ESCs. For initial testing, Nyx was operated in a "wheels up" position with the main goal being verification of functionality and testing failsafes. Initial experimentation went quickly and without issue. The next phase was stress testing. This meant that the RageBridge would need to be installed in the robot, which presented two problems. The first problem was the large number of exposed contacts that would be ill

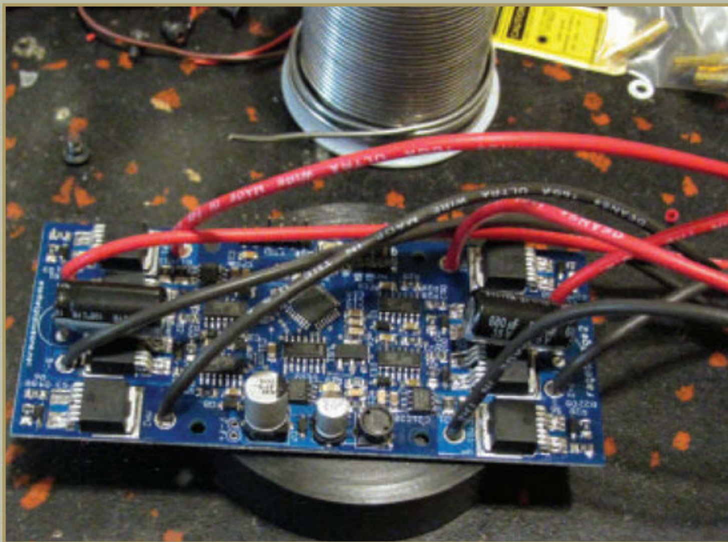


FIGURE 2. All of the wires soldered on.

FIGURE 3. The RageBridge wrapped up to protect it from debris and minimize the risk of short circuits.

served by being shorted. The second was that there was very little extra room in Nyx for a long speed controller.

The first problem was relatively easy to solve. I had a supply of dry vinyl tape on hand. This tape doesn't have an adhesive but sticks to itself reasonably well. This meant no residue on the RageBridge, plus easy removal should I need to access the board. With that solved, it was time to wedge it into the chassis.

With the RageBridge installed and all of the controls acting as desired, it was time for stress testing. The experiments in this situation would involve driving the robot in a manner that replicated many worst case scenario maneuvers commonly seen in robot combat. These maneuvers included hard forward to reverse transitions, aggressive turning, rapid acceleration, and random direction changes. The tests ran until the battery was low on power or until failure.

Two testing videos are posted on my YouTube channel (**www.youtube.com/user/mikencr**) and are titled "RageBridge Testing-Nyx" and "RageBridge Testing Part 2-Nyx."

There was an issue during initial testing where the robot would occasionally become briefly unresponsive during highly aggressive maneuvering. After working with Equals Zero Designs, the problem was eventually traced to both the RageBridge and my old motor controllers having battery elimination circuits (one was still powering the weapon on Nyx) that were outputting differing voltages. When the BEC on the weapon motor controller was disabled, performance improved greatly (as is seen in the second test video).

After running extensive trials on the RageBridge, I am very happy with the controller. Compared to my previous solution, the throttle response feels better, and even with current limiting there isn't a noticeable drop in speed or acceleration rate. For robots with a tight weight budget, it's also very light compared to many of its competitors.

I'm convinced the RageBridge is a great option for motor control in 12-30 lb robots, and I have seen them used in 60 lb robots without issue. SV

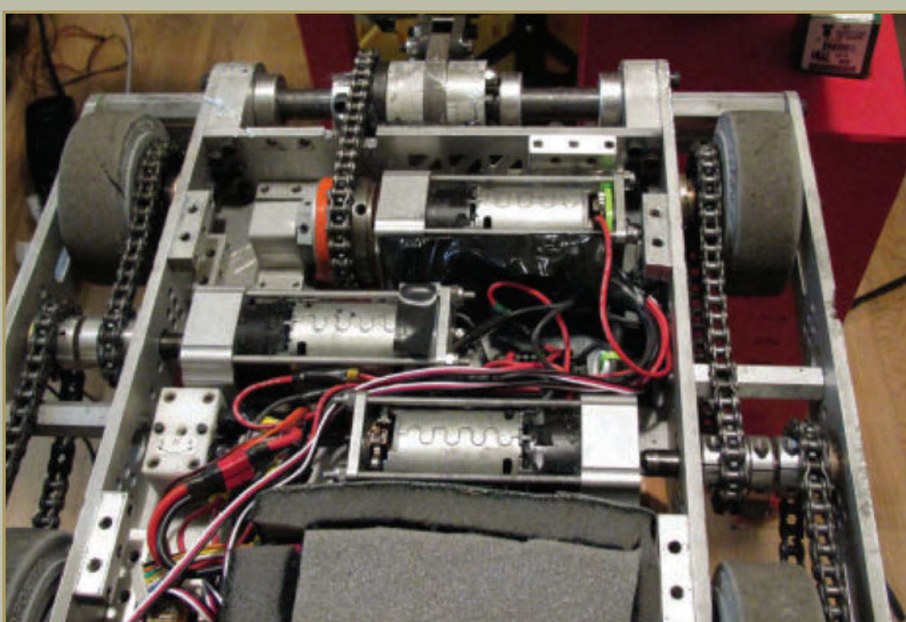


FIGURE 4. One of the only locations where the RageBridge would fit in any orientation.

PARTS IS PARTS:

Prototype Testing: FingerTech Robotics Mini-Spark

● by Mike Jeffries

In the 150 g weight class, there are only a few options for drive systems. FingerTech Robotics is looking to add some diversity to the market with their new Mini-Spark gearboxes.

The Mini-Sparks will come in a range of ratios (10, 20, 30, 50, 150, 210, and 298:1), but for this test I opted for the 20:1 ratio as that would result in very good speed with a two cell LiPo and the 1.4" wheels I was intending to use.

Externally, the Mini-Spark resembles many of the competitor motors. However, there are a few characteristics that separate it from the other motors in the same weight range.

First, there are several gear ratios to choose from that result in usable rpm ranges. Second, the hardened gears mean the gearbox will likely survive more abuse which can be the difference in a match.

Third, the motor is slightly heavier and noticeably more powerful than the motors used on similar gearboxes.

The small size and common mount points make them a potential swap for existing gearboxes. For this test, however, I built a new robot that was designed to stress the gearboxes.

With highly visible wheels and a horizontal spinning weapon, the gearboxes are exposed to shock loads not only from other robots, but from its own weapon throwing the whole robot around the arena.

The first round of testing was purely functional, and consisted of basic driving maneuvers in an open area to verify that there were no defects in the gearboxes. After that, the next round of testing was aggressive maneuvering which consisted of quick direction changes, slamming the robot into the wall, and bouncing the wheels off of the

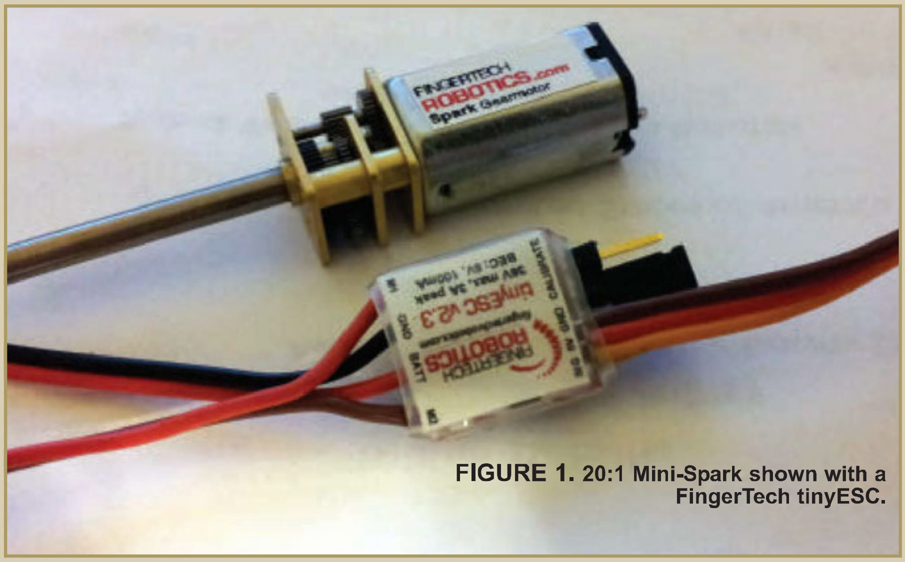


FIGURE 1. 20:1 Mini-Spark shown with a FingerTech tinyESC.

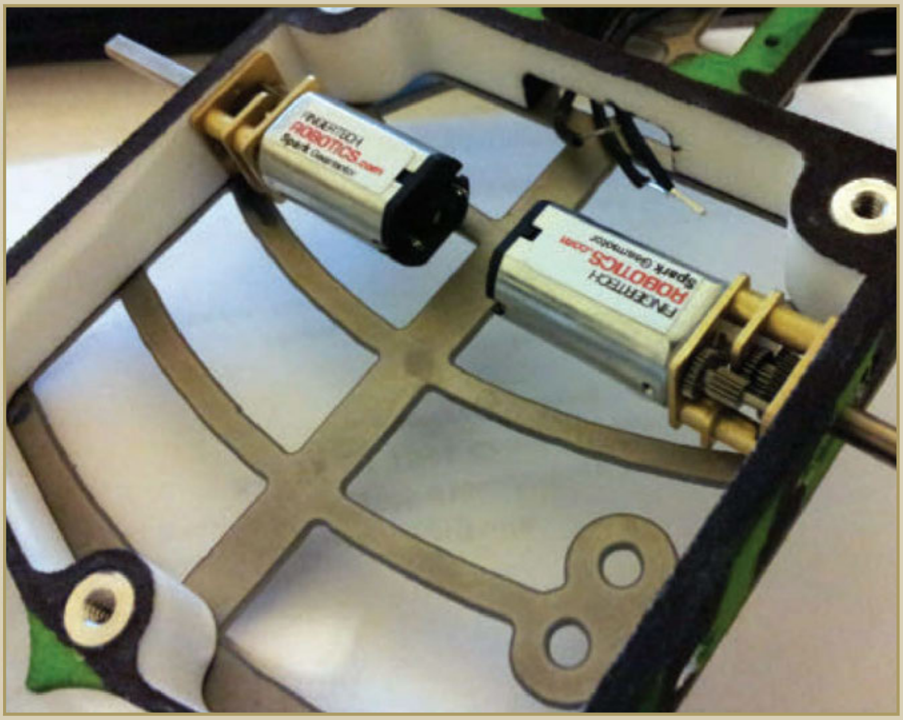


FIGURE 2. Two Mini-Sparks mounted in the chassis of the bot that is being used for testing.

FIGURE 3. Even with the small size of the Mini-Sparks, space is at a premium in a 150 g combat robot.

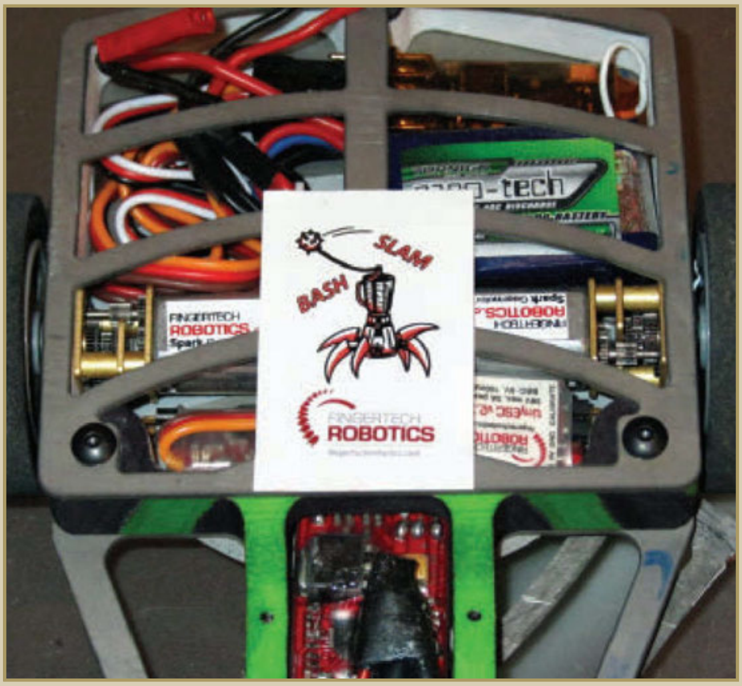


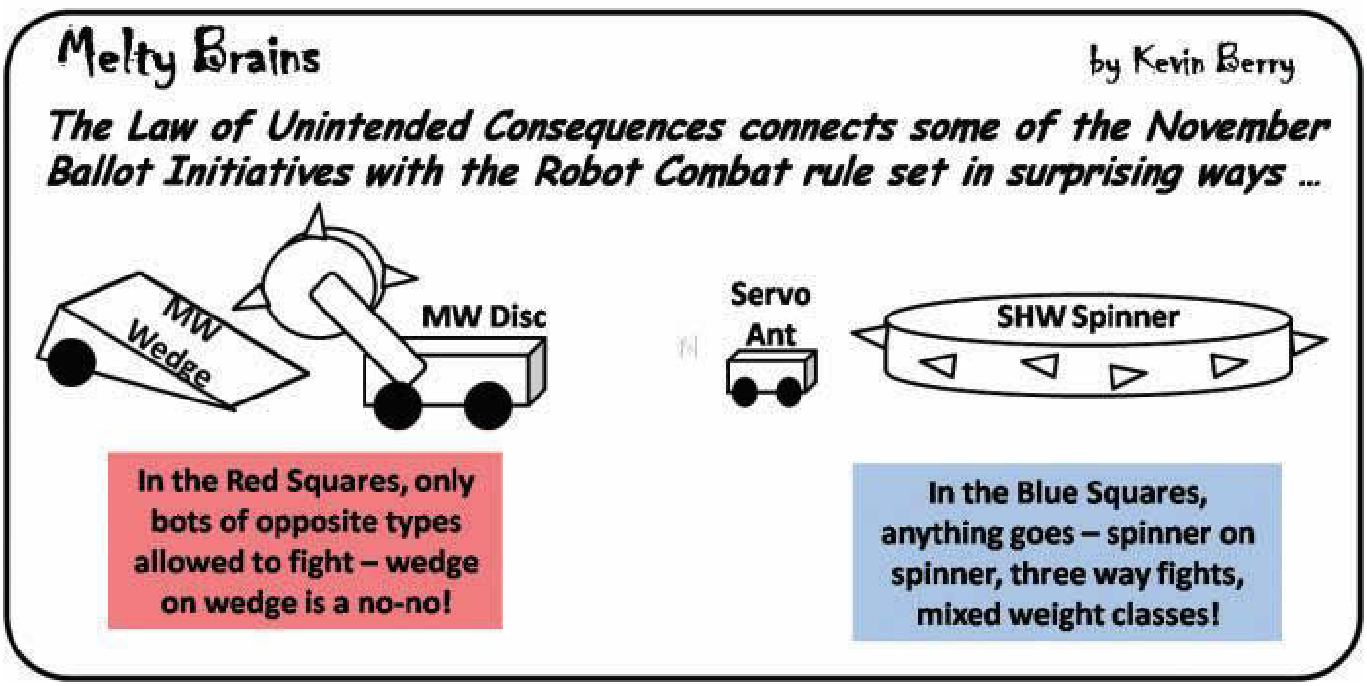
FIGURE 4. There is no visible wear in the gearboxes after all of the tests.

wall as best I could.

The final stage of testing was using the weapon to hit objects, which resulted in the robot sending itself in a random direction with a decent amount of energy.

All of these tests have been completed successfully and can be viewed on my YouTube page (**www.youtube.com/user/mikencr**) under the titles "Keres Drive System Test," "Keres/FingerTech Mini-Spark Testing," "Keres Weapon Test," and "Keres Weapon Test Part 2."]

With the tests completed, I have to say I am happy with the new Mini-Spark gearboxes and believe they are a great drive system option for 150 g combat robots. Based on the performance, I believe they are also a strong option for weaponed 1 lb battling bots. SV



Building a Large Scrolling Display — Part 1: Getting One Panel Working

● by Pete Smith

I often attend shows where I'm either demonstrating combat robots and bot hockey, or as a sponsor providing prizes of Kitbots products. In order to inform the public about the event or the products, I have several banners to hang up on the walls, plus I also hand out business cards and information sheets when I have the time. It would be useful, however, to have a more eye-catching and informative way of passing on the information.

I'd noticed scrolling LED displays doing just that in shop windows, etc., and even priced them on eBay. They were all pretty expensive, though — especially in the larger sizes. Then, one day I noticed an advertisement for the Freetronics Dot Matrix Display in a magazine.

The Freetronics Dot Matrix Display or DMD (**Figures 1** and **2**) has 512 LEDS in a 32x16 matrix which is mounted on a PCB (printed circuit board) and protected by a plastic frame — all for about $100 delivered. The display can be controlled using an Arduino, so for about $150 in total one could have a 6" high scrolling display that could be programmed using a PC to display a message about as long as you could desire.

I'm a mechanical engineer and while I have dabbled with the odd electronics project over the years, I have no experience with small computer boards like the Arduino. One thing that I liked about the Freetronics DMD is that they also make a suitable Arduino board — the "Eleven" — and have excellent beginner's instructions available on their website at **www.freetronics.com.**

I ordered a single DMD and an "Eleven" microcontroller board; it arrived in a couple of weeks from Australia (they are also available from resellers in the US).

My first job was to get the microcontroller board working. I followed the simple instructions provided, and downloaded the latest Arduino IDE and the required USB driver. Both installed easily on my Windows 7 laptop. The IDE is the software that is required to write, edit, and compile programs

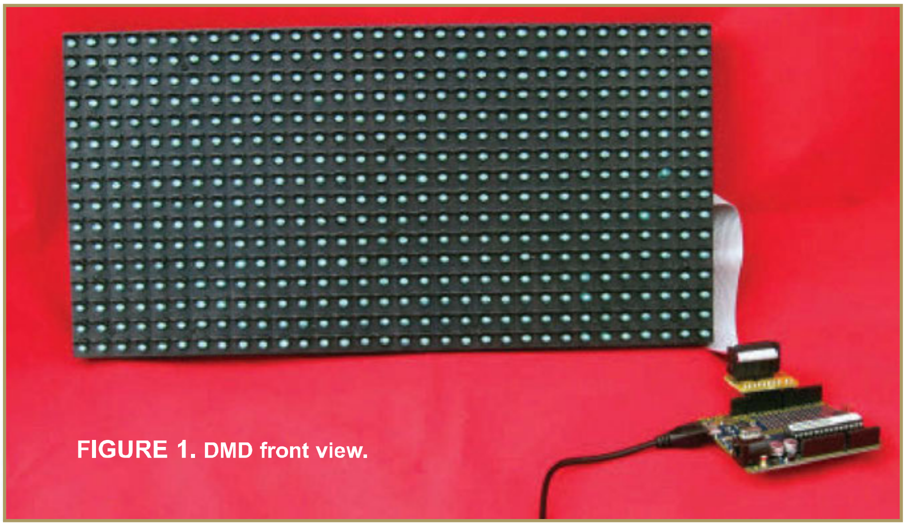


FIGURE 1. DMD front view.

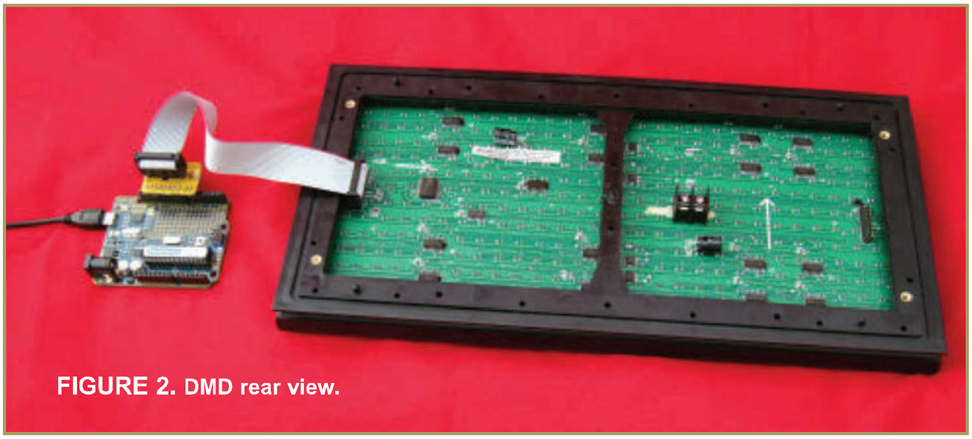


FIGURE 2. DMD rear view.

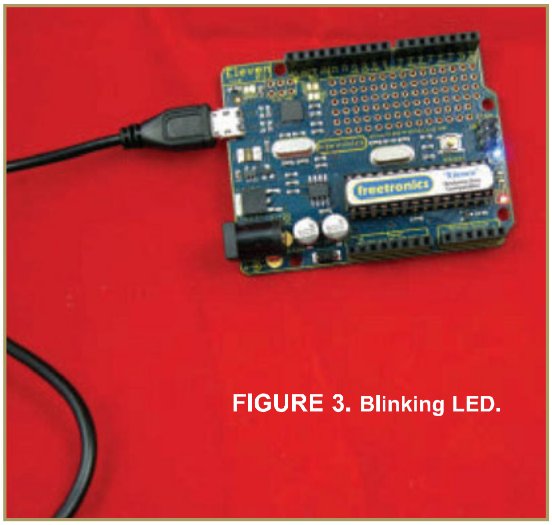


FIGURE 3. Blinking LED.

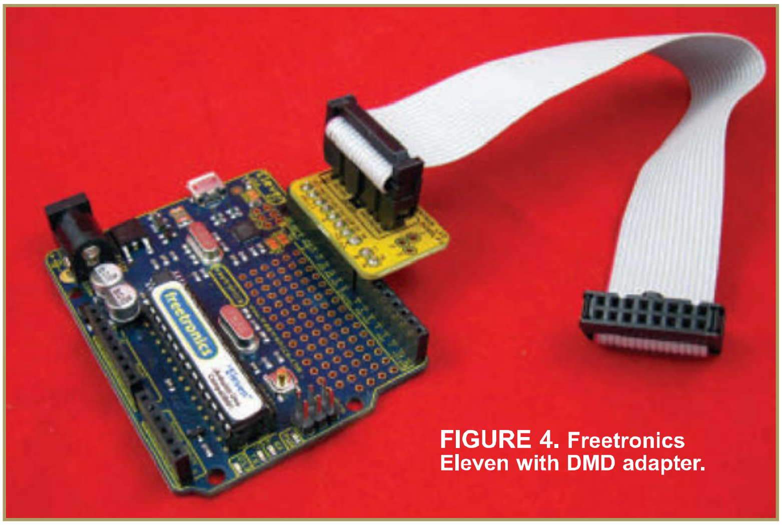


FIGURE 4. Freetronics Eleven with DMD adapter.

to run on your Arduino, and the driver is necessary to get your computer to recognize the Eleven so that you can download the programs on to it. An included USB cable provides the power and data connection to your PC.

The Eleven comes with a preloaded "Blink" program (they are called sketches). So, once everything is installed correctly and you connect the Eleven to your computer, you can see that it's working correctly with a blue LED power indication and a red LED that blinks on and off (**Figure 3**).

You can then quickly test the ability to upload a new sketch by following the clear instructions to load the Blink sketch that comes as an example with the IDE, and then play around with changing values in the sketch to make the LED flash on and then off at various speeds. I found this was all pretty easy and was feeling pretty confident when I moved on to attach the DMD.

The DMD is connected to the microcontroller using the provided adapter board and ribbon cable. This simply plugs in as shown in **Figure 4**. The cable supplies both power and data to the DMD. A trial sketch (*DMDscrolltext.pde*) is available for download from the Freetronics website. However, this is where I ran into difficulties.

I downloaded the sketch and tried to compile it, but all I got was an error code. It was getting late by the time I found this problem, so I decided to try again in the morning. My mind was obviously working on the problem overnight, as in the morning it suddenly occurred to me that one had to have the sketch located in the same place in the file structure as the Blink sketch had been. I created a new folder in "Examples" called "9.DMD," then another folder within that called "DMDscrolltext," and placed the *DMDscrolltext.pde* file within that folder.

I then compiled and uploaded the sketch to the DMD and it worked perfectly! The display is surprisingly bright — almost painfully so — at close range. This is not particularly obvious in photos or even videos, but it's certainly bright enough for an indoor venue.

Next, I played around with editing the demo program so that only the scrolling text section remained, and then altered it as a first pass at what I was trying to achieve. The results can be seen at **www.youtube.com/watch?v=qO2SANz2Y8s.**

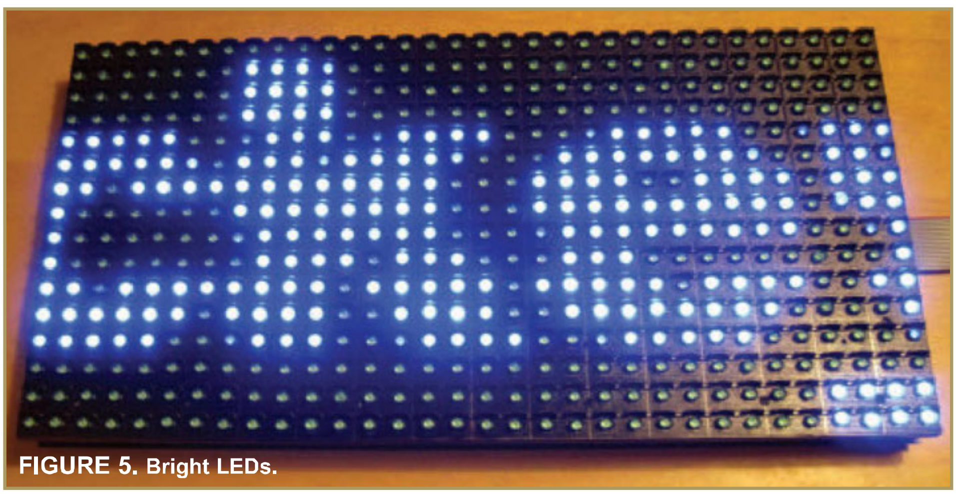
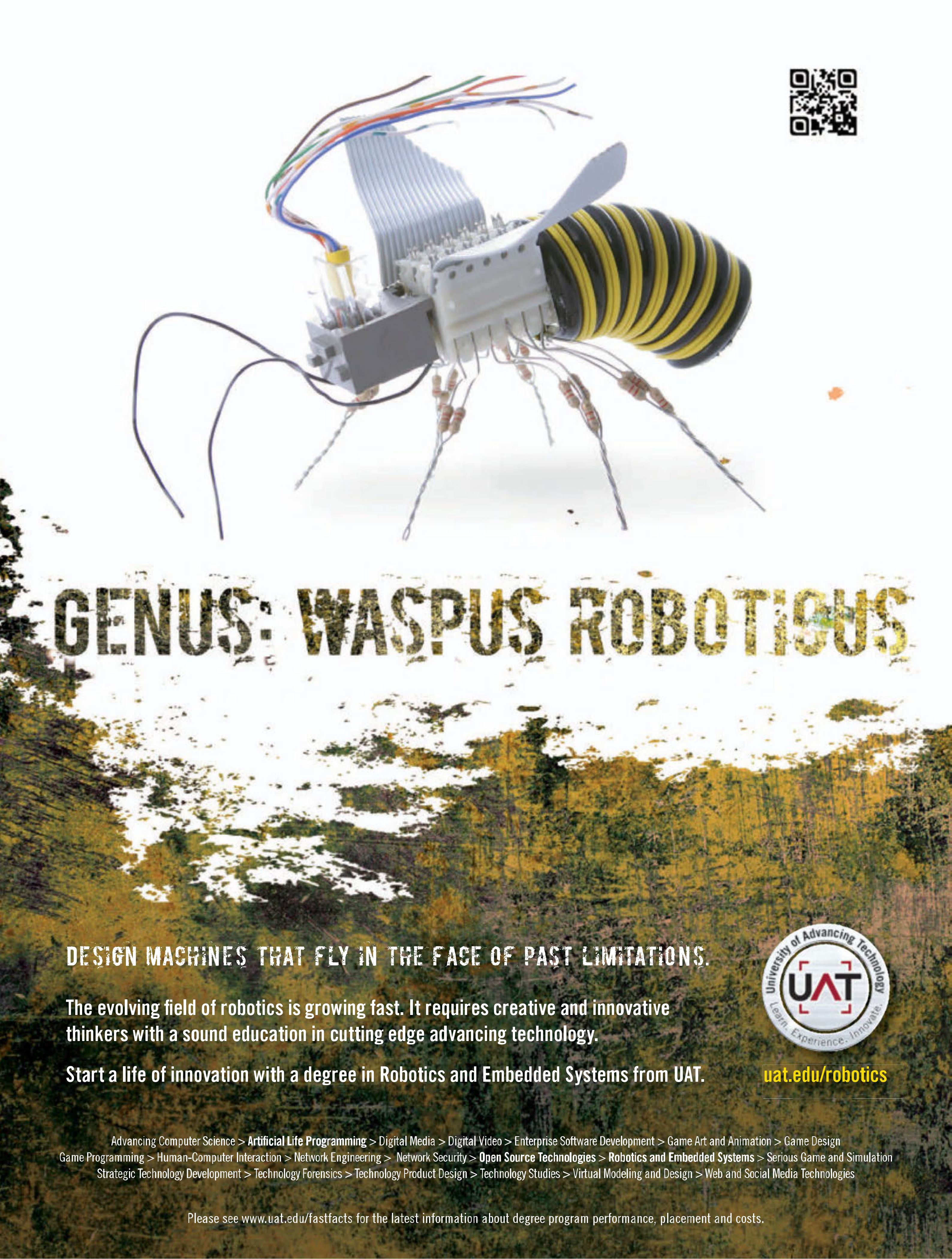


FIGURE 5. Bright LEDs.

Apart from the problem with the file structure, both the Freetronics Eleven and DMD proved easy to get up and running. However, I do need to get more familiar with how the Arduino coding works if I really want to get the display to do all it is capable of.

In Part 2, I will show you how I added a second DMD and an external power supply so that the display could run without being attached to my computer. **SV**


GENUS: WASPUS ROBOTICUS
DESIGN MACHINES THAT FLY IN THE FACE OF PAST LIMITATIONS.
The evolving field of robotics is growing fast. It requires creative and innovative thinkers with a sound education in cutting edge advancing technology.
Start a life of innovation with a degree in Robotics and Embedded Systems from UAT.
University of Advancing Technology
UAT
Learn. Experience. Innovate.
uat.edu/robotics
Advancing Computer Science > Artificial Life Programming > Digital Media > Digital Video > Enterprise Software Development > Game Art and Animation > Game Design
Game Programming > Human-Computer Interaction > Network Engineering > Network Security > Open Source Technologies > Robotics and Embedded Systems > Serious Game and Simulation
Strategic Technology Development > Technology Forensics > Technology Product Design > Technology Studies > Virtual Modeling and Design > Web and Social Media Technologies
Please see www.uat.edu/fastfacts for the latest information about degree program performance, placement and costs.

Broadcasting Real Time Video With Your Robot

by Gordon McComb

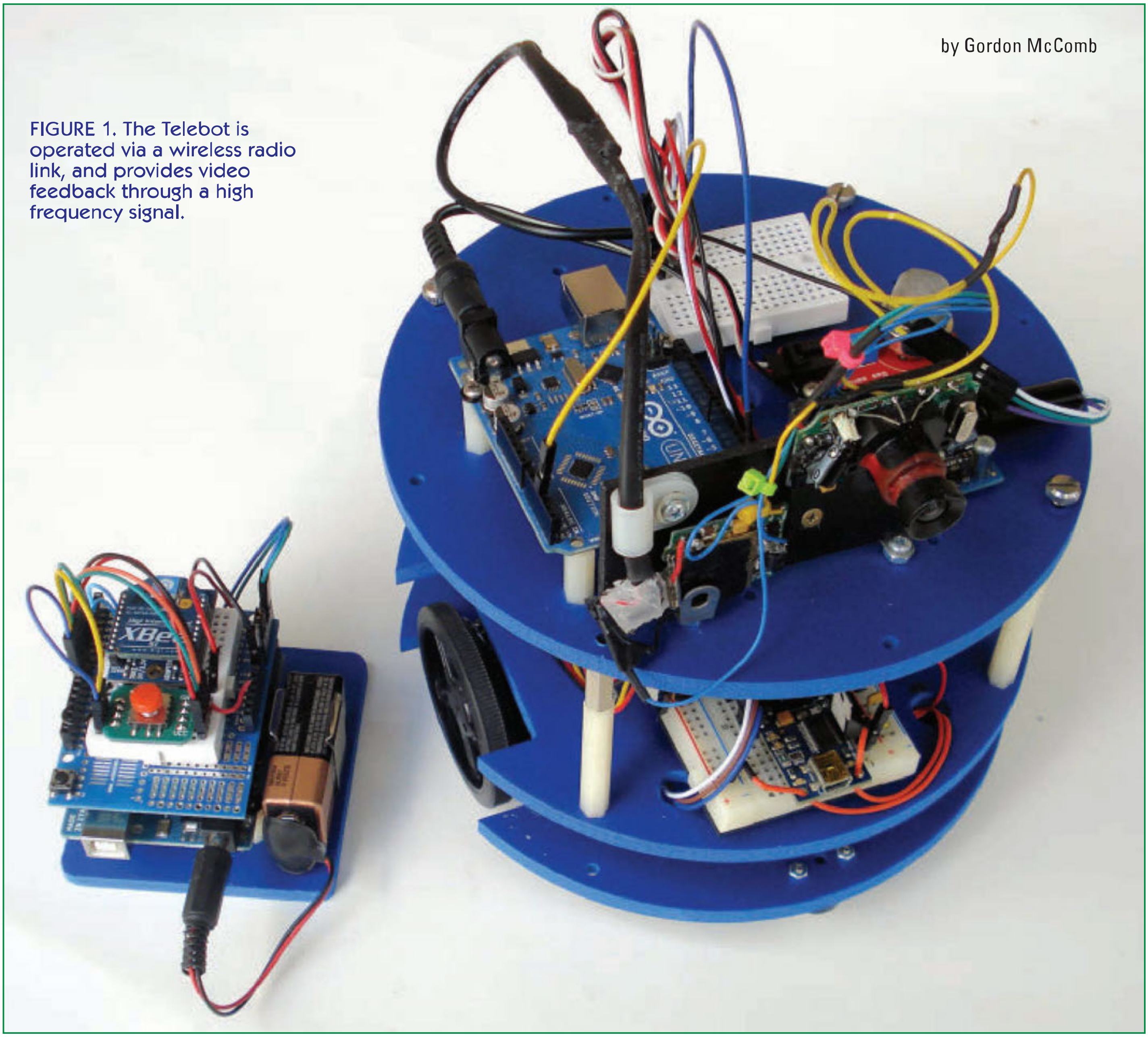


FIGURE 1. The Telebot is operated via a wireless radio link, and provides video feedback through a high frequency signal.

Back in the November 2012 issue, I wrote about the Telebot — an Arduino-based deskbot operated via XBee wireless control. Telebot's main operations are controlled by a human, but it's not a simple stop-and-go contraption. Low level functions are handled by the onboard Arduino microcontroller. The robot can react to touch and distance sensors, automatically avoiding obstacles for you.

Let's go even further with the Telebot. In addition to controlling the robot remotely, how about having the Telebot send back video pictures of the landscape it sees. Wireless security cameras are relatively low cost — some under $40, including both camera/transmitter and receiver — and easy to use on a mobile robot. The camera is operated using a battery or other convenient power supply.

This month, I'll describe how to add video to any robot — remotely controlled or not. In addition to simply feeding back a picture of whatever is in your bot's path, I'll also cover how to overlay textual data onto the video, so you can get real time telemetry of your machine's current status.

My prototype Telebot is shown in **Figure 1**; you're free to adapt the ideas to whatever robot design you currently have, or plan to develop.

The Basics of Robot Video

Figure 2 shows a small wireless camera along with its paired receiver. These setups are available for under $75 — sometimes less — and are easy to use. The camera is operated by its own nine volt battery, while the receiver uses a plug-in power adapter. The video output of the receiver connects to any compatible monitor.

I'm using an NTSC camera and receiver, so any NTSC-compatible monitor will do. A small 7" LCD monitor is ideal, since these have a power jack on them for low voltage (usually 12 volts or less) operation. Rig up a battery for both the receiver and monitor, and you can create a fully portable video reception unit.

The standard frequency band used by wireless video cameras is 2.4 GHz, which just so happens to be the same band used by your XBee radios. There is a chance that the video camera will interfere with the XBee, and vice versa.

However, this problem is mitigated by using a multi-channel wireless camera where you can select a different RF frequency within the 2.4 GHz band.

You can tell if your camera and XBee are messing each other up by monitoring the video image when you place the XBee remote next to the camera. If there's a problem,

FIGURE 2. You can add video to your robot with a low cost camera and receiver pair, like this one. The camera contains its own transmitter and antenna, and is designed for battery operation.

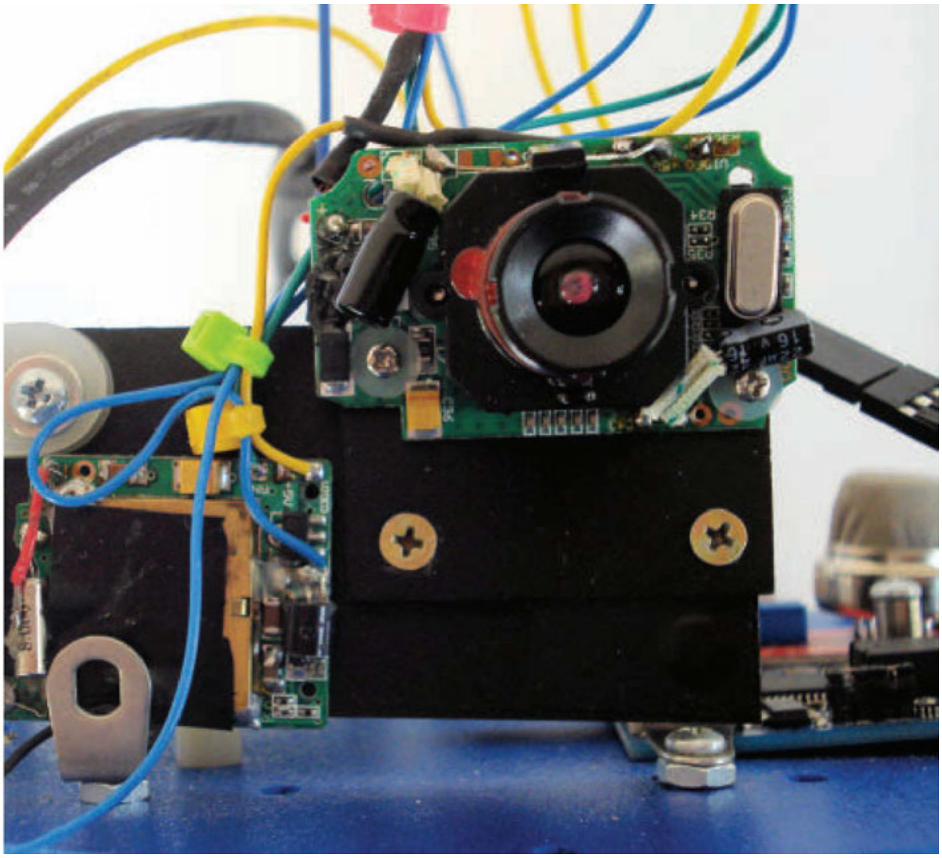


FIGURE 3. Hacked backup camera, showing the separate transmitter and camera. It's ugly, but it works. (I don't advise you to try the same; it seems backup cameras use quite a bit of weatherproofing, making them undesirable hacking candidates.)

you'll see wavy lines and other interference patterns.

If your wireless camera lacks a channel changing feature or if selecting none of its channels removes the interference, you can try altering the XBee's channel; there are 16 to choose from, and each channel occupies its own 5 MHz space. The channel on both the transmitter and receiver XBee must be tuned to the same channel.

Refer to the documentation on the XBee radios on how to set channels. It can be done directly in the Arduino sketch (using something called *Command Mode*), though most people prefer to use the separately available X-TCU program which provides a handy graphical interface. You need an XBee carrier board with USB in order to communicate between your PC and the radio. The X-TCU program is available for download from **digi.com** — makers of the XBee.

Completely self-contained cameras (like the one in **Figure 2**) are useful for video-only broadcasting. The camera unit contains the camera itself plus the broadcast circuitry, and even the antenna. They're less useful when you want to superimpose text over the video. For that, you'll either need a separate camera and transmitter pair, or you'll have to hack an all-in-one camera to tap into the video signal before it gets to the transmitter circuitry.

Hacking is the approach I took for my Telebot, and it wasn't easy. **Figure 3** shows a close-up of a severely hacked up backup camera mounted on the third deck of the Telebot. The second and third decks are separated by long standoffs.

Using a bit of sleuthing, I was able to cut into the video stream and route it to the circuit (described in the next section) that superimposes text. These cameras sometimes use the standardized wiring colors of red for power, black for ground, and yellow for video. If your camera does, you're in luck; otherwise, you'll need to use a meter — or better yet, an oscilloscope — to find the proper wires.

It turns out backup cameras are not the best choice

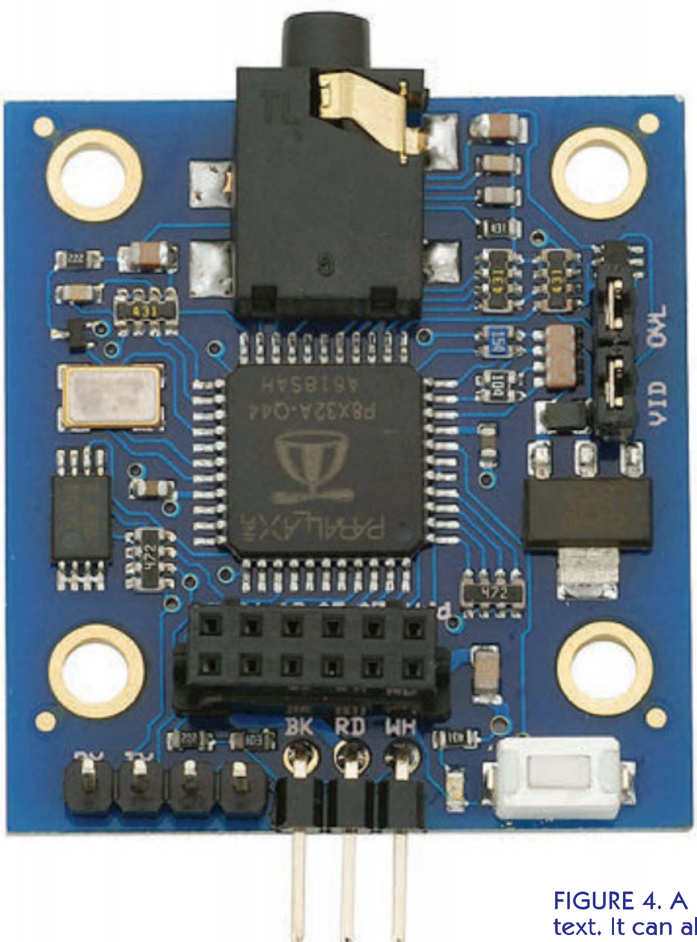


FIGURE 4. A Parallax Backpack module is designed to generate video text. It can also be used to overlay text onto the signal from a video camera. *Photo courtesy of Parallax, Inc.*

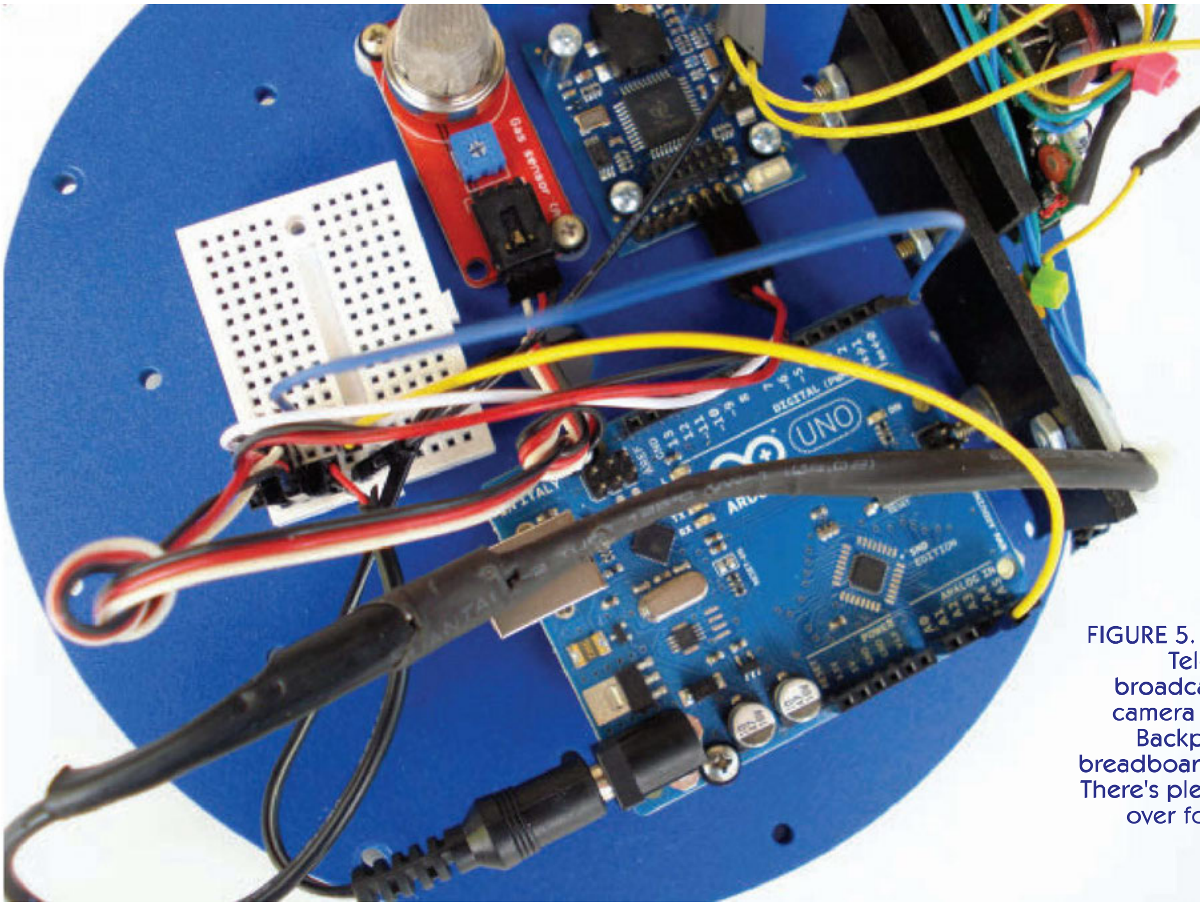


FIGURE 5. Top view of the Telebot with video broadcast, showing the camera and transmitter, Backpack, gas sensor, breadboard, and Arduino. There's plenty of room left over for more sensors.

for this work, as most are — or should be — sealed against the elements. The sealing compound proved difficult to remove, which was made worse because of the miniaturization of all the components. If you decide to hack a self-contained camera-transmitter unit, I suggest you try a standard indoor model.

How to Superimpose Text Over Video

Not only can your bot beam back video, you can superimpose text and other images onto the video — giving a kind of Terminator-style view of the world around it. The technique involves passing the video through an *overlay* device — an electronic circuit that's capable of locking onto the incoming video signal and superimposing text over it.

While you can use an Arduino to overlay text onto video, the process involves added circuitry, and the heavy demands of video processing leaves the Arduino little time to do anything else. So instead, I decided to use a Propeller Backpack (available from **parallax.com**).

The Backpack (shown in **Figure 4**) is a video generating and processing add-on that connects to most any microcontroller. You simply attach a video source to the input, route the output to a wireless transmitter or monitor, and use simple serial commands to place text anywhere over the image. You have two choices as to where you insert the Backpack in the video stream. Take your pick:

- **Between the camera and transmitter.** This works best if your wireless system has a separate camera and transmitter, and both are reasonably high quality. If your wireless camera is all-in-one (as noted earlier), you will need to tap into the video line to make the connection to the Backpack. Not impossible, but depending on the camera, such electronic "surgery" can get tricky. You might even wreck the camera attempting it, so be careful.

- **Between the transmitter and the monitor.** In order for this to work, you need to use the XBee radio on the Telebot to send back data to your base station. You can use the Arduino on the remote control to receive the data, but you'll need to add a bigger breadboard for the Backpack, or else mount the Backpack elsewhere. The Backpack and Arduino are connected via short wires. (One idea: Put the Backpack under the Arduino.)

For the Telebot prototype, I elected to put the Backpack between the camera and transmitter. After tapping through to separate the video wire, I mounted the camera (with lens) and the transmitter on a piece of 1/8" plastic, then attached that using brackets to the third deck of the Telebot. The video input from the camera connects to the Backpack which then connects to the transmitter, as shown in **Figure 5**.

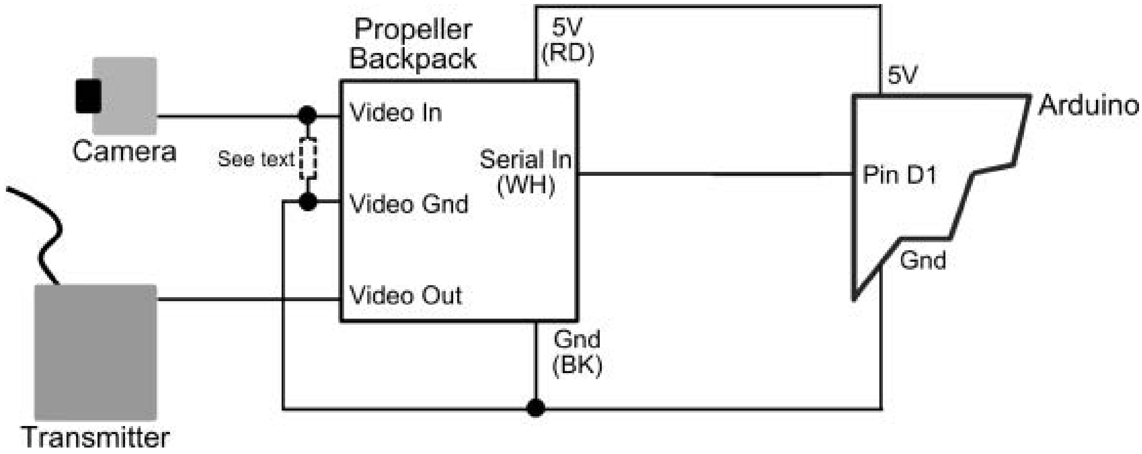


FIGURE 6. Wiring diagram for the Propeller Backpack video overlay module.

Figure 6 shows the connection diagram between the camera, Backpack, and Arduino. You'll need to make a connector for piping the video through the Backpack. Use a 1/8" stereo plug, and wire the video in, video out, and ground connections as shown in **Figure 7**. Keep wire lengths short; the shorter, the better to reduce signal noise and interference. If you have it, three-conductor shielded cable (two inner conductors, one outer shield) will provide better immunity to noise. The outer shield is connected to ground.

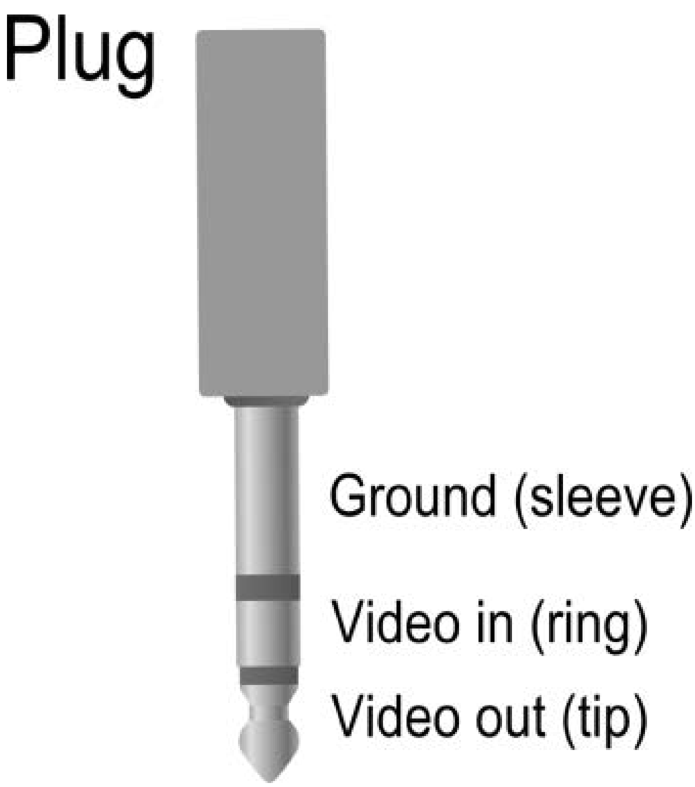


FIGURE 7. Use a 1/8" stereo plug to connect the video in/video out signals to the Backpack.

Not all cameras output perfect video. The Backpack requires a fairly standard video signal, or else it may not be able to properly overlay text. If the level of the video signal is too high, the text may not appear, or may appear overly dark or light.

If that happens, try placing a small 1/8 watt resistor across the camera input. Experiment with the value of the resistor, but a good starting point is 150 Ω.

Using the Propeller Backpack

Before you can use the Backpack, you must first reprogram it to work in overlay mode (it comes pre-programmed for displaying text over a black or colored background). In order to reprogram the Backpack, you'll need a Prop Plug (also from Parallax) to connect your PC to the Propeller via USB. Fetch and install the Parallax Propeller Tool software from the Parallax site, and locate the overlay programming code — it's provided as a link on the Backpack page. Use this code in the Propeller Tool software to upload the new programming to the Backpack.

Once reprogrammed, you can mount the Backpack to the third deck of the Telebot. You only need to use one I/O line for basic communications with the Backpack. I elected to use the Arduino's hardware serial port to connect to the Backpack. Only the D1 transmit (Tx) pin is connected; the D0 receive (Rx) pin is left unattached.

You can keep the Backpack connected to the Arduino while uploading sketches to it. The Backpack will also receive the programming data, but will be unable to make much sense of it. In some cases, the intercepted data may cause the Backpack to lock up. If this occurs, then after the upload is complete, momentarily disconnect power and let the Arduino and Backpack reset.

To write text to the Backpack, you first set up what its documentation refers to as *displays* and *windows*. Once these have been defined, you can write text into the display. The sketch in **Listing 1** shows a Hello World example of preparing the Backpack, then displaying a line

Listing 1. Telebot_Backpack

```
// For Arduino 1.0 IDE or later

const int backPack = 1;
        // Pin D1 for Backpack serial
//Define constants for Backpack
#define APNDSP      0x18
#define CHGCLR      0x12
#define CLS         0x00
#define DEFWIN      0x10
#define HOME        0x01
#define LF          0x0a
#define MOVWIN      0x19
#define SHODSP      0x07
#define SHOWIN      0x1a
#define USEWIN      0x11
#define WDWRAP      0x15

#define DBL         0x80
#define SHO         0x40
#define TPT         0x01

void setup() {
  // Reset Backpack
  pinMode(backPack, OUTPUT);
  digitalWrite(backPack, LOW);
  delay(100);
  pinMode(backPack, INPUT);
  delay(2000);
  // End reset Backpack

  Serial.begin(9600);
  delay(1000);

  byte windowA = 1;
  byte displayNum = 2;
  defineWindow(windowA, 44, 2);
        // Define window 44 x 2
  changeColor();
  appendDisplay(windowA, displayNum, 1, 1);
  Serial.println("This text displayed on the
  Backpack");
  delay(2000);
        // Wait 3 seconds
  showDisplay(0);
        // Hide text
}

// Empty loop; for this demo everything is done
// in setup
void loop() { }

// Backpack helper functions follow

// Define a window, given a window number,
//   number of columns, number of rows
void defineWindow(byte windowNum, byte cols,
byte rows) {
  Serial.write(DEFWIN);
  Serial.write(windowNum);
  Serial.write(cols);
  Serial.write(rows);
}

// "Append" a window to a display, given a
// window number
//   display number, and x and y coordinates on
// the screen

void appendDisplay(byte windowNum,
byte displayNum, byte x, byte y) {
  Serial.write(APNDSP);
      // Append display to window
  Serial.write(displayNum);         // display
  Serial.write(SHO | windowNum);    // winno
  Serial.write(x);                  // x
  Serial.write(y);                  // y
  Serial.write(SHODSP);
      // Show display
  Serial.write(displayNum);
  Serial.write(WDWRAP);
      // Turn on word wrap
  Serial.write(1);
}

// Change color of text and background
void changeColor() {
  Serial.write(CHGCLR);
  Serial.write(0x0f);              // mask
  Serial.write(0x08);              // trans
  Serial.write(0x08);              // foreground
  Serial.write(0x00);              // background
}

// Move the window to a new coordinate on the
// screen
void moveWindow(byte displayNum, byte x, byte y)
{
  Serial.write(MOVWIN);
  Serial.write(displayNum);
  Serial.write(x);
  Serial.write(y);
}

// Use (activate) a window
void useWindow(byte windowNum) {
  Serial.write(USEWIN);
  Serial.write(windowNum);
}

// Show (or hide) a display (shows or hides the
// text)
void showDisplay(byte displayNum) {
  Serial.write(SHODSP);
  Serial.write(displayNum);
}

// Clears all text
void cls() {
  Serial.write(CLS);
}
```

of text. To make it easier to use the Backpack, I've created several functions that encapsulate common tasks.

Important! Be sure to *thoroughly* read the Propeller Backpack Video Overlay Object documentation — especially the General Concepts section — which describes displays, windows, and slots. You'll especially want to study how the Backpack keeps track of the current window and display, and how commands subsequently sent affect the current settings.

Also note that (as of this writing) the Backpack firmware does not support two side-by-side windows for text. You can stack windows vertically, but you can't place them beside one another.

In operation, the sketch begins by first resetting the Backpack. This is done by making pin D1 an OUTPUT, and bringing the pin low for 100 milliseconds. The pin is then made an INPUT, so that it can be used by the Arduino hardware serial port. Then, and only then, is the hardware

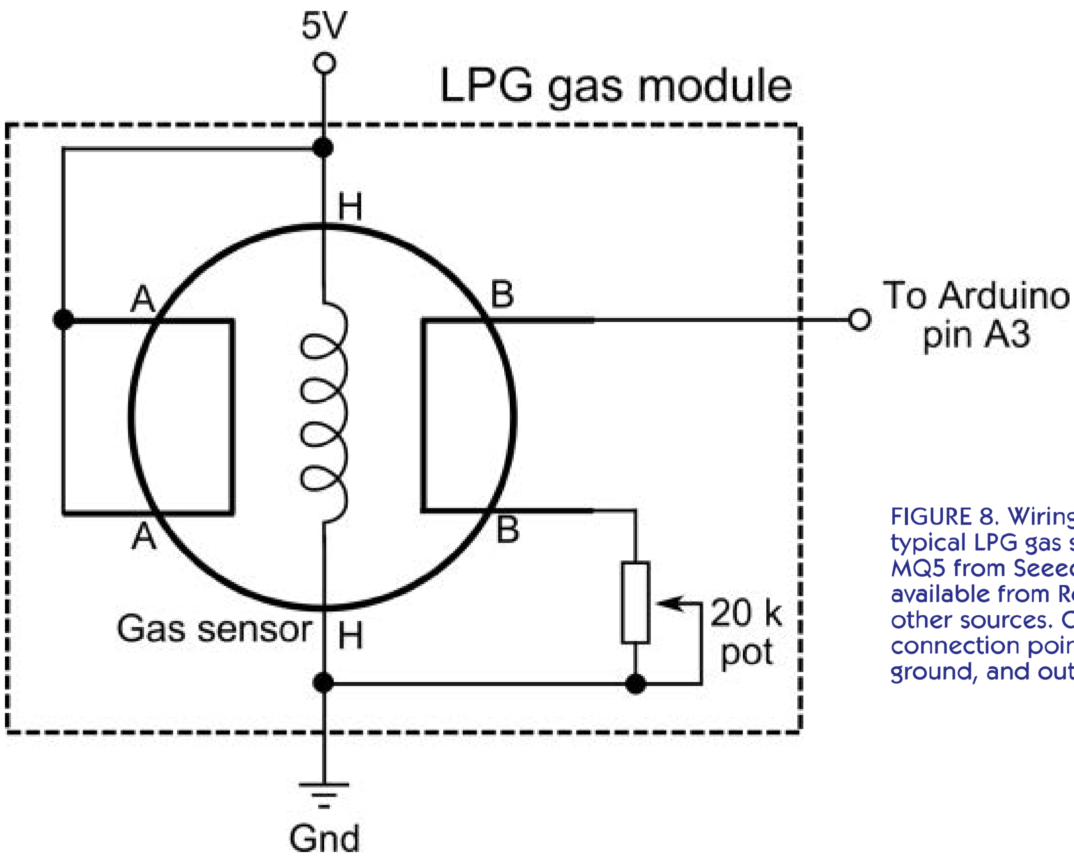


FIGURE 8. Wiring diagram for a typical LPG gas sensor; this one is an MQ5 from Seeed Studio, and is available from RobotShop, among other sources. Convenient connection points provide for 5V, ground, and output.

serial port set up to transmit data to the Backpack.

From here, the sketch sets up a *window* and a *display* for holding a line of text. The numeric values you assign here are arbitrary; I'm using 1 and 2, respectively. The functions *defineWindow*, *changeColor*, and *appendDisplay* are ones I've defined; the routines themselves are at the bottom of the sketch. The functions merely send out bytes that prepare the Backpack display.

For example, the *defineWindow* function sends four bytes to the Backpack:

```
Serial.write(DEFWIN);
       // Define window command (0x10)
Serial.write(windowNum);
       // Window number (e.g. 1)
Serial.write(cols);
       // Number of text columns
Serial.write(rows);
       // Number of text rows
```

Displaying Environment Data

Any sensor that provides printable text can be used with the Telebot to obtain environmental data. As a demonstration, I'll show the use of an LPG sensor to detect levels of natural gas. The sensor is easily connected to the Arduino (see **Figure 8**) and returns an analog value indicating the concentration of natural and other gasses.

Important note: The LPG sensor demonstrated here is for *educational use* only. Don't trust it as the sole noxious gas detector in your home. Since natural gas is lighter than air, it will rise to the ceiling and concentrate there — away from a low-lying robot like the Telebot.

Other kinds of environmental sensors you might try include different types of gas sensors (alcohol, carbon dioxide, etc.), humidity, temperature, barometric pressure, compass heading, light level, and so on. The following code snippet assumes you've already set up the Backpack to display text. The *Serial.print* lines send data to the Backpack, since it's connected to the Arduino's hardware serial transmit pin:

```
void loop() {
  cls();                        // Clear old display
  val = analogRead(A3);         // Read sensor
  Serial.print("Gas level: "):
  Serial.println(val, DEC); // Print sensor value
  delay(1000);                  // Wait 1 second
}
```

For my prototype, I used an MQ5 LPG gas sensor module from Seeed Studio. This module is self-contained, providing connection points for 5V power, ground, and analog output. An onboard potentiometer allows you to adjust the sensitivity of the device. On initial use, dial the pot to its mid-point position, and adjust as needed.

Enhancements for Your Tele-robot

Before closing, I'd like to suggest a few neat features you might like to add to your Telebot. But don't just stop here — let your imagination soar!

Add a speaker to the remote control unit to provide audible feedback. Use a piezo element for the loudest sound and least amount of current draw. Avoid using an 8 Ω dynamic speaker; it'll draw too much current. You can cut down on the current demand by installing a 100-150 Ω resistor in series between the speaker and Arduino, but the sound may not be very loud.

Mount a vibrator motor to the base plate, and interface it to the Arduino using a transistor driver (the ULN2003 driver chip is ideal for this; it contains seven separate drivers). Add some collision detectors to the Telebot, and when one is activated, use the XBee link to transmit an "I hit something!" signal. Briefly apply power to the motor when the remote gets the bump command.

Use a Microsoft Kinect connected to your PC to transmit motion commands to the Telebot. You can do away with the Arduino on the remote, and instead connect the XBee transmitter (you'll need an XBee USB carrier) to your computer. The Microsoft Robotics Developer Studio (RDS) provides a programming platform for interfacing with and using the Kinect.

For more ideas on building robots using the Arduino, check out my new book, *Arduino Robot Bonanza*, due in April. It contains the full Telebot project including many additional enhancements, plus five more ready-to-build robot plans. **SV**

Sources

Budget Robotics
ArdBot/Telebot chassis kit
www.budgetrobotics.com

Computer Geeks (among many others)
Low cost wireless camera kits
www.geeks.com

Parallax
Propeller Backpack board, gas sensors
www.parallax.com

RobotShop
LPG gas sensor
(other gas sensors are also available)
www.robotshop.com

Build the Kronos Flyer

Part 4: Setup

by Michael Simpson

www.servomagazine.com/index.php?/magazine/article/february2013_Simpson

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

In this article, you will program and calibrate your ESCs (electronic speed controllers); you will set the direction of your motors; I will show you set-by-step how to install your NAZA flight controller; you will install your props; and, by the time you finish, you will be ready to take your Kronos Flyer for a spin.

I won't be covering the KK2 flight controller in this article, so if you decided to use this one, you will have to visit my website for tips on installation and setup.

ESC Setup

Programming the ESCs

Don't panic! You don't really have to program the ESCs. The program card simply allows you to change the settings on them. Both the Plush and MultiStar ESCs are set by default to LiPo (lithium-ion polymer) batteries. The problem with this setting is that the ESCs will cut off when the battery voltage reaches a certain level. This is done to keep from damaging the LiPo battery. A fixed wing plane can still be landed with the ESC cut off; a multi-rotor will flip and crash when one or more of its ESCs cut off.

To fix this, we simply need to tell the ESCs we are using a NiMh (nickel-metal hydride) battery, even though we are not. This is the main setting we must change; the other settings are just tweaks.

First, Let's Look at Programming a Plush ESC

Start by plugging one of the ESCs into the programming card as shown in **Figure 1**. Notice the black lead on the ESC cable is mated with the lead marked with the negative symbol. The white lead is mated with the lead marked Signal. Make sure the battery is disconnected when you attach the ESC to the card.

Plug the battery into the battery connector on the power distribution board as shown in **Figure 2**.

It will take a second or two, and then the card will light up as shown in **Figure 3**. The LEDs shown are the default settings for the ESCs. To change a setting, move the cursor to the row you want to change by pressing the up/down arrow button (first button). The cursor is represented by a flashing LED. To change the setting, use the right/left arrow button (second button). Once you are happy with the setting, move to the next row. When you are satisfied with all the settings, hit the OK button. Once complete, your settings should look like the ones in **Figure 4**.

The Plush settings are as follows:

- Brake: Off
- Battery Type: Ni-xx
- Cut-off Type: Soft
- Cut-off Voltage: Low
- Start Mode: Normal
- Timing Mode: Middle
- Music: All Off
- Governor: Off

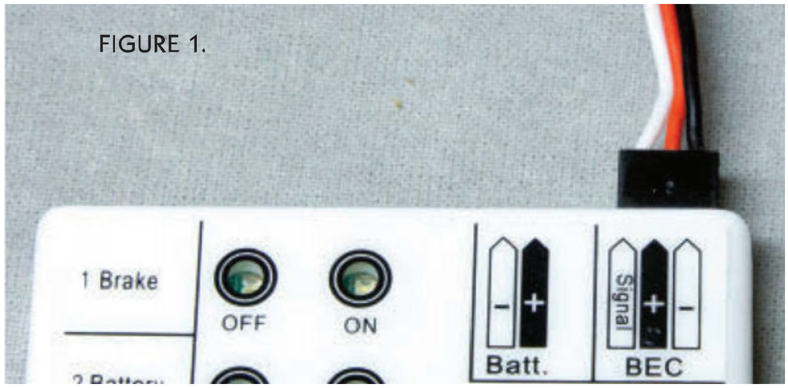


FIGURE 1.

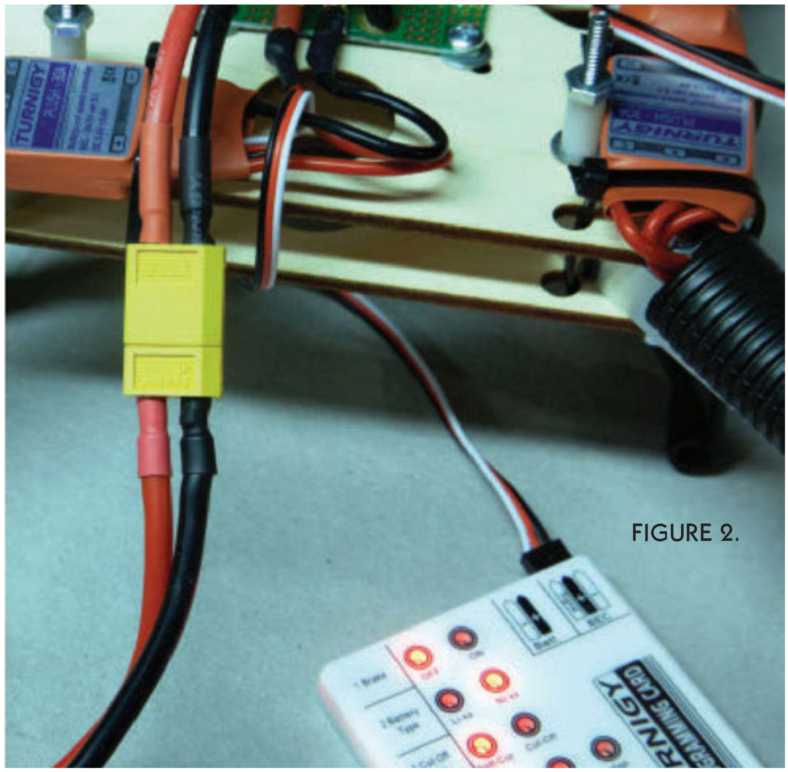


FIGURE 2.

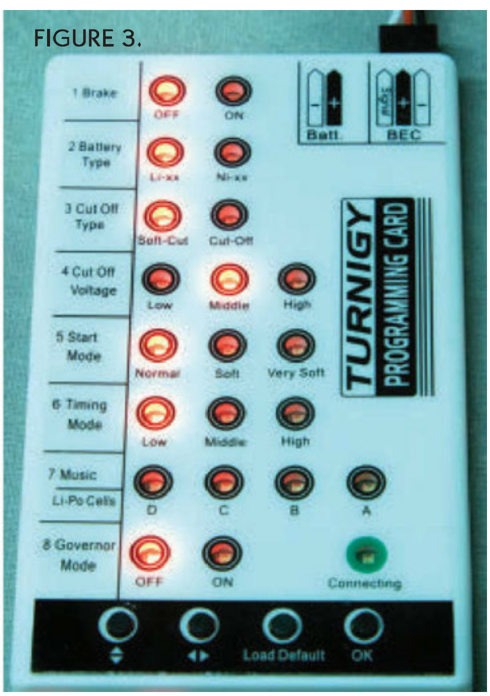


FIGURE 3.

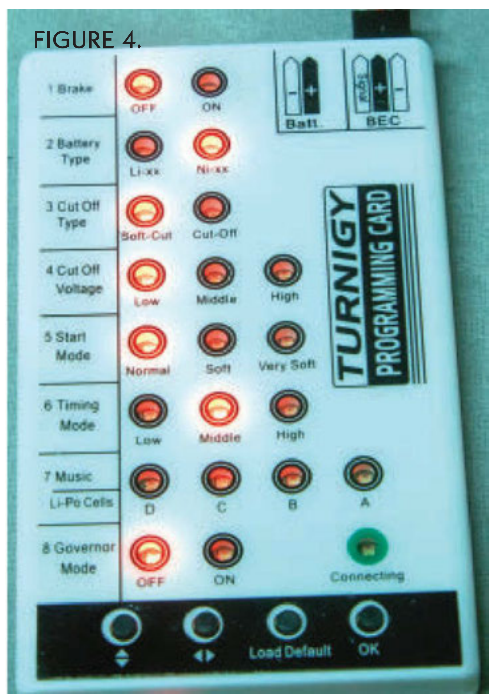


FIGURE 4.

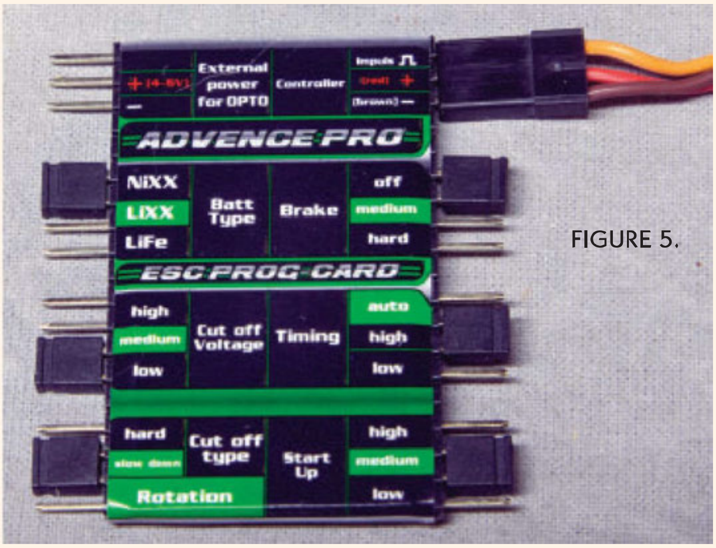


FIGURE 5.

Programming a MultiStar ESC

The MultiStar ESCs are a little easier to program. You simply set the jumpers on the card as shown in **Figure 5**. Plug the ESC cable into the card as shown, then attach the battery and wait for the motor to beep. Then, remove the battery and move to the next ESC.

The MultiStar settings are as follows:

- Brake: Off
- Battery Type: Ni-xx
- Cut-off Type: Slow Down
- Cut-off Voltage: Low
- Start Up: Medium
- Timing Mode: Auto

Calibrating the ESCs

Calibrating the ESCs is very easy. Please note that you should not have the props attached when you are doing this.

First, plug the ESC cable into the out jack on the servo tester. Be sure to observe the polarity of the cable. Move the knob on the tester fully clockwise (max) as shown in

Motor Setup

Don't put your servo tester away just yet. You need to use it to test the direction of your motors. Connect the servo tester to ESC #1 using **Figure 8** as a reference. Make sure the servo tester is adjusted so that it is at its minimum setting (full CCW). Attach the battery.

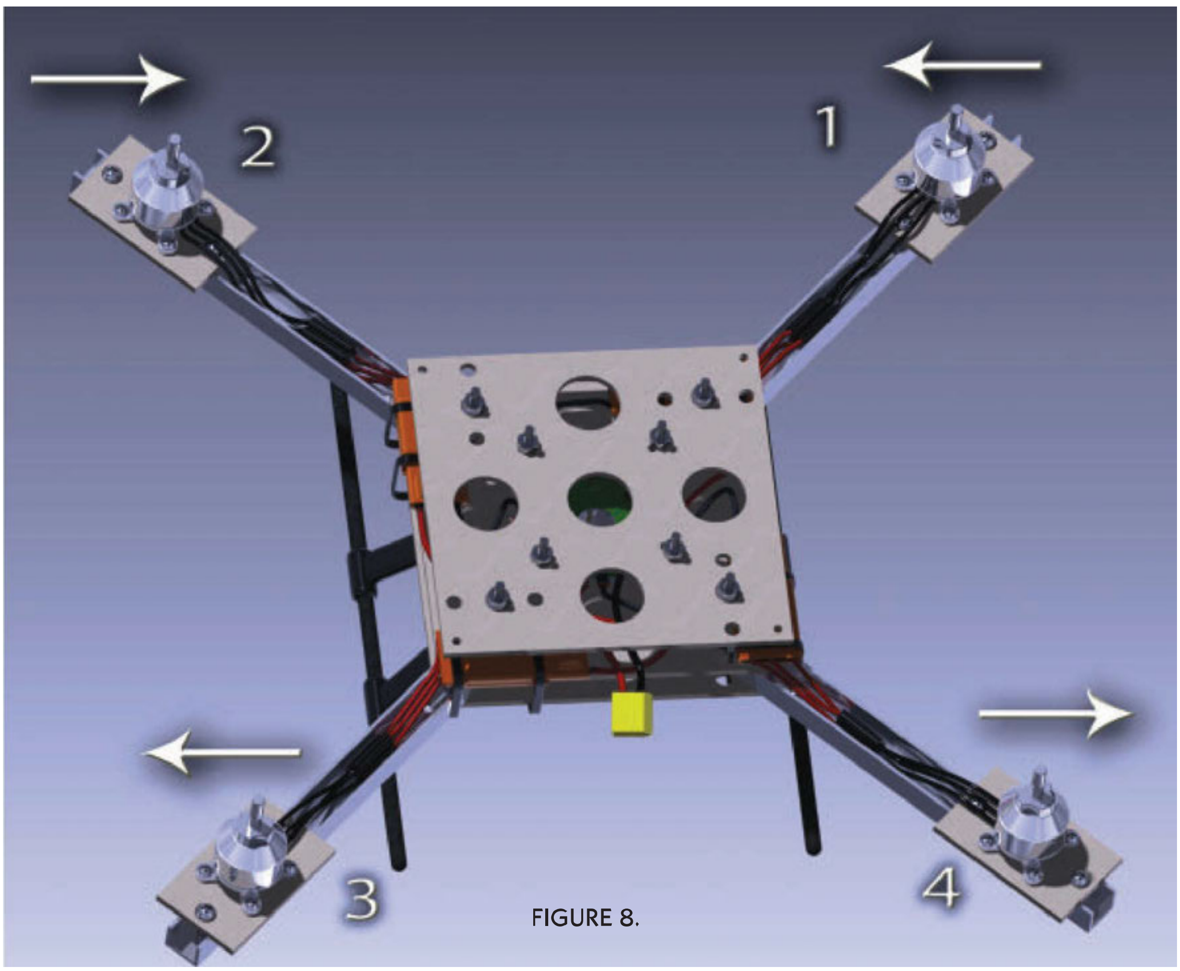


FIGURE 8.

Figure 6.

Attach the battery. When you hear the motor beep twice, spin the knob fully counterclockwise as shown in **Figure 7**. The motor should beep once. The ESC is now calibrated.

Continue to do all the ESCs this way. You need to disconnect the power before each calibration.

FIGURE 6.

FIGURE 7.

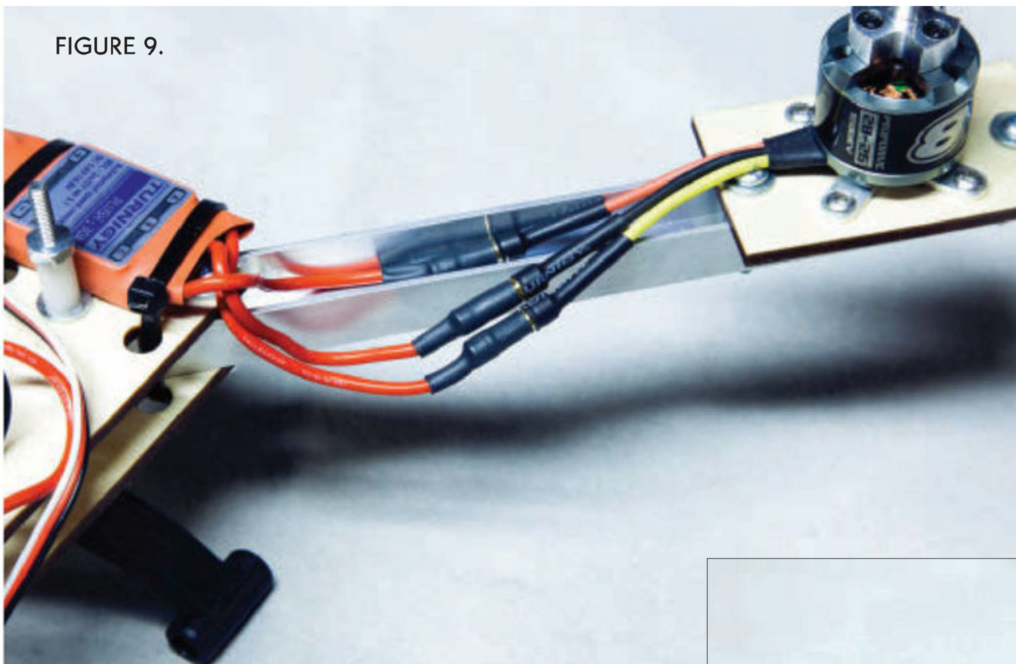


FIGURE 9.

Now, slowly turn the servo tester clockwise until the motor starts turning. Notice the arrows in **Figure 8**. They indicate the direction that motor #1 should be turning. If it's not, disconnect the battery and swap two of the wires between the motor and the ESC as in **Figure 9**. Any two will do. This will reverse the direction of the motor. Repeat the test.

Do this for all four motors/ESC combinations, noting the direction of the arrow in **Figure 8**.

Once you have your motors set up, you can add a piece of split loom cable to cover the wires. The cable shown in **Figure 10** is 1/2" and held in place with small tie wraps. Note that this is only an option and not required.

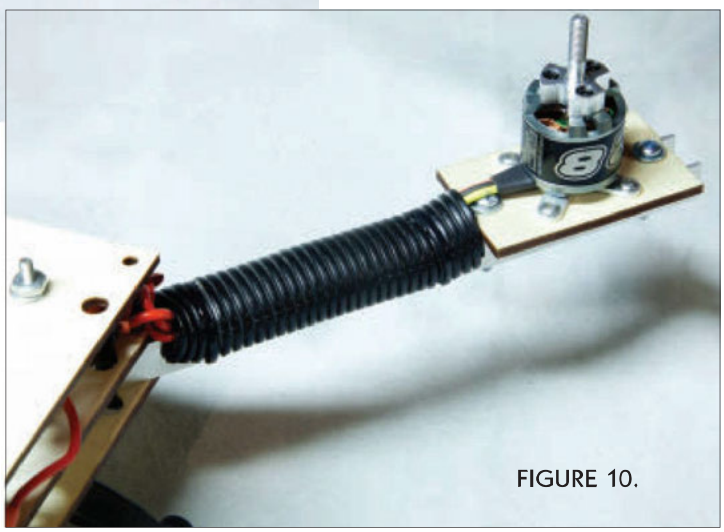


FIGURE 10.

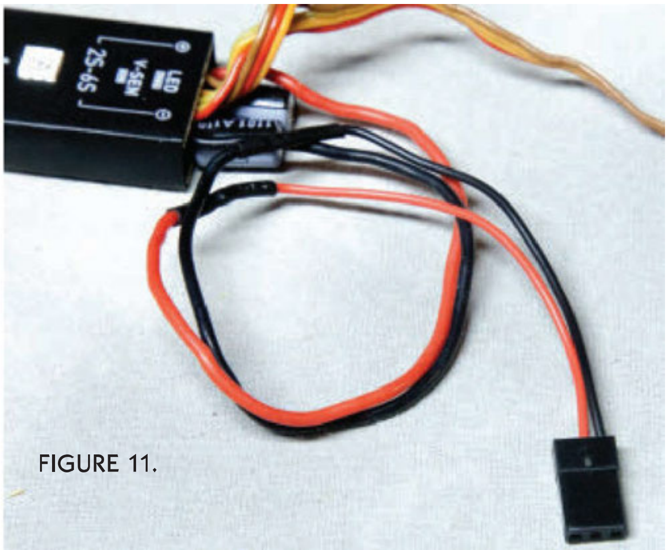


FIGURE 11.

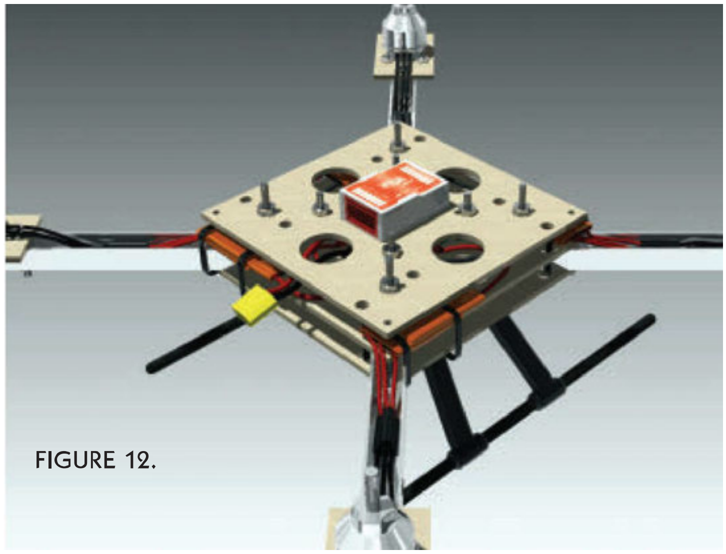


FIGURE 12.

NAZA Hookup

The motor directions are set, and the ESCs are programmed and calibrated. Before we attach the props, let's install the NAZA flight controller.

VU Prep

The NAZA VU (Versatile Unit) has to be connected directly to your battery. The VU is what powers your NAZA and radio receiver. It also has the ability to monitor the battery level. Unfortunately, the VU does not come with a connector. The easiest way to get the VU connected to your battery is to cut a servo extender in half, and solder the red and black leads on the servo connector to the red and black leads on the VU as shown in **Figure 11**. Remove the unused signal wire from the connector.

Attach the NAZA

Add some double-stick foam tape to the bottom of the NAZA unit and attach it to the center of the component platform as shown in **Figure 12**. The end with the M1, M2, M3 connections should be facing the front. The end of the Kronos Flyer with the battery connector is positioned at the rear of the craft.

Unlike other flight controllers, the NAZA has built-in vibration dampening, so it can be attached directly to the platform.

Attach the VU

Add some double-stick foam tape to the bottom of the VU, and attach it in the location shown in **Figure 13**. Leave enough room to run the VU wires through the large hole.

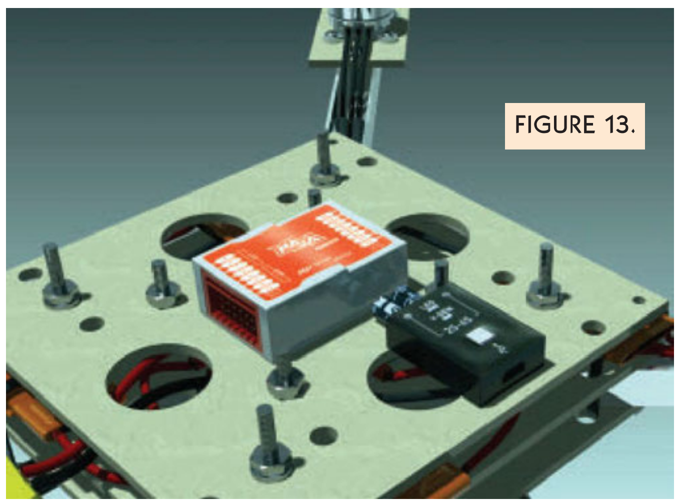


FIGURE 13.

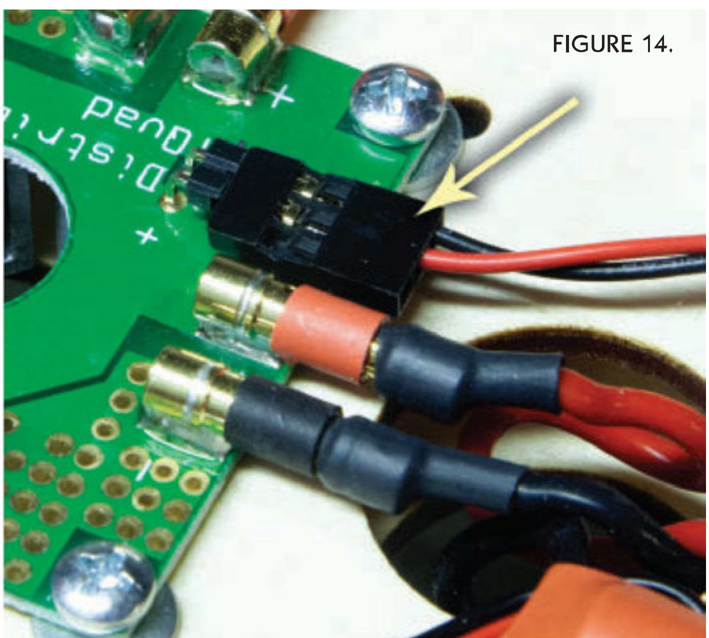


FIGURE 14.

Remove the four nuts holding the component platform in place, and route the VU wires through the large hole at its side. Plug the connector into the small two-pin header on the distribution board as shown in **Figure 14**. Be sure to observe polarity. The pin on the red wire should be mated to the terminal marked with +. The pin on the black wire should be mated to the other terminal. Reattach the component platform.

Attach Receiver

Place some double-stick foam tape on the bottom of your receiver and attach it as shown in **Figure 15**. Here, I am showing a Futaba receiver, but there should be plenty of room for almost any other type. Make sure the receiver's connectors are facing the rear of the craft.

Attach ESC Cables

Route the four ESC control cables through the large hole in the front of the component platform. Referring back to **Figure 8**, attach the ESC control cable that is connected to motor #1 to the terminals on the NAZA marked M1. The black wire on the ESC connector should be pointed up as shown in **Figure 16**.

Repeat the process for all the ESC control cables, matching the motor number to the same number on the NAZA. The unused connections on the NAZA are used for other configurations.

Route the antenna wires on the receiver through one of the holes in the platform. This is done to keep the props from hitting the wires.

Important!!!

Double-check your cables. If you switch one of them, your quad will flip the first time you try to fly it.

Attach Receiver Cables

Attach your receiver cables to the NAZA as shown in **Figure 17**. For the Futaba 8FG, use **Table 1** as a guide for your connections. If you are using a Turnigy 9X for your radio, use **Table 2** for your connections. For other radios, you will have to do research, but chances are one of these connection tables will work for you.

The NAZA comes with all the cables needed to connect the receiver to the NAZA. They are Futaba style cables, and will work with both the Futaba and Turnigy radios.

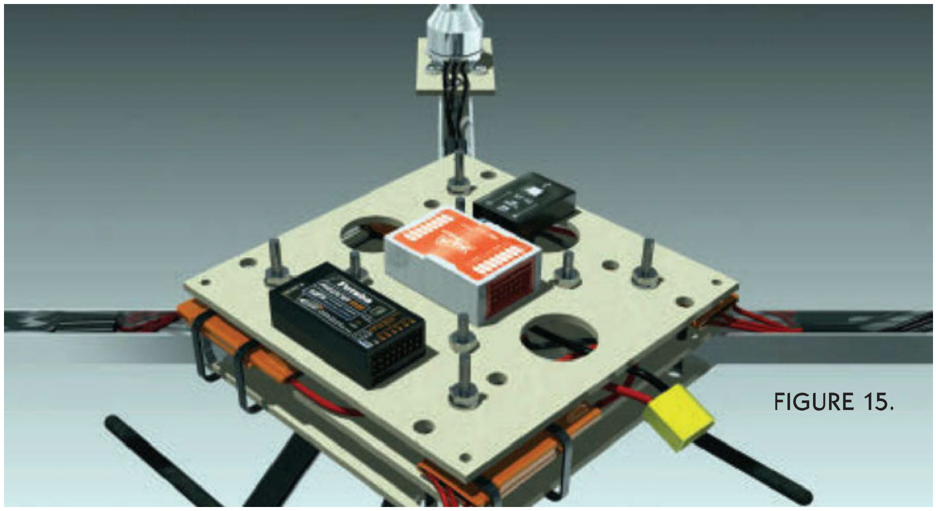


FIGURE 15.

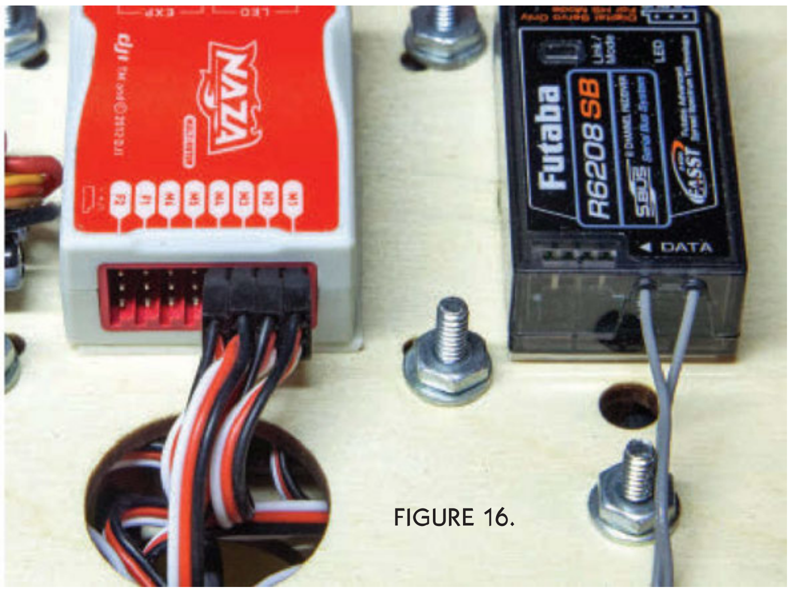


FIGURE 16.

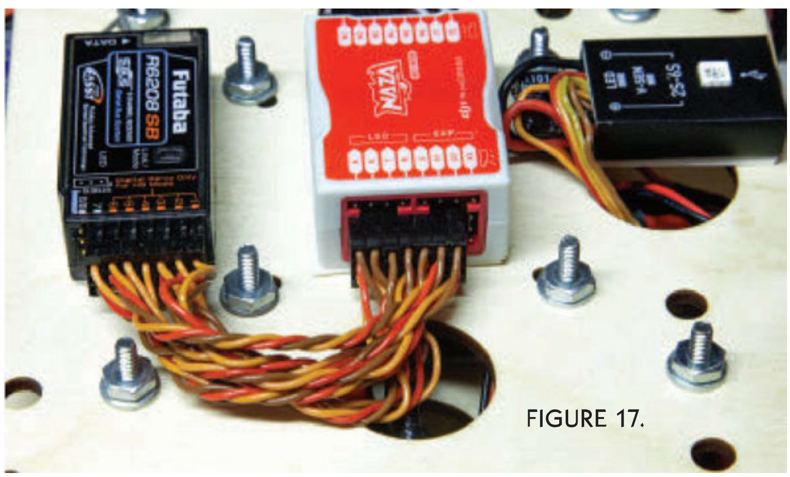


FIGURE 17.

Receiver Channel	NAZA Connector
1	A
2	E
3	T
4	R
5	U
6	X1
7	X2

Table 1 (Futaba 8FG).

Receiver Channel	NAZA Connector
1	A
2	E
3	T
4	R
5	-
6	U
7	X1
8	X2

Table 2 (Turnigy 9X).

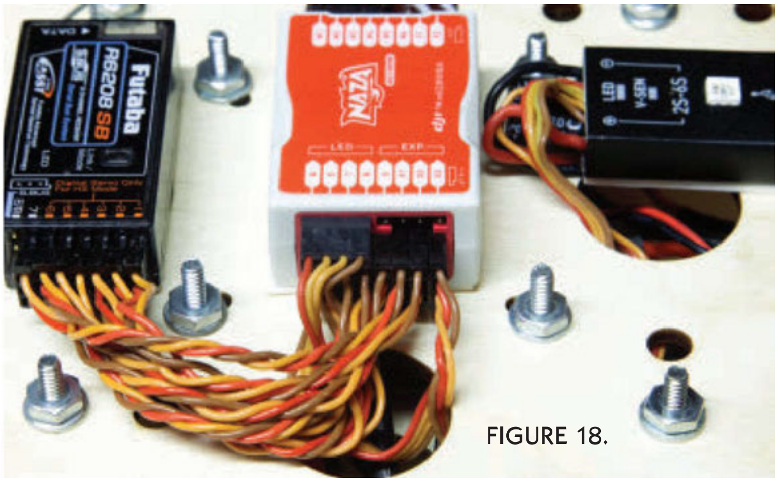


FIGURE 18.

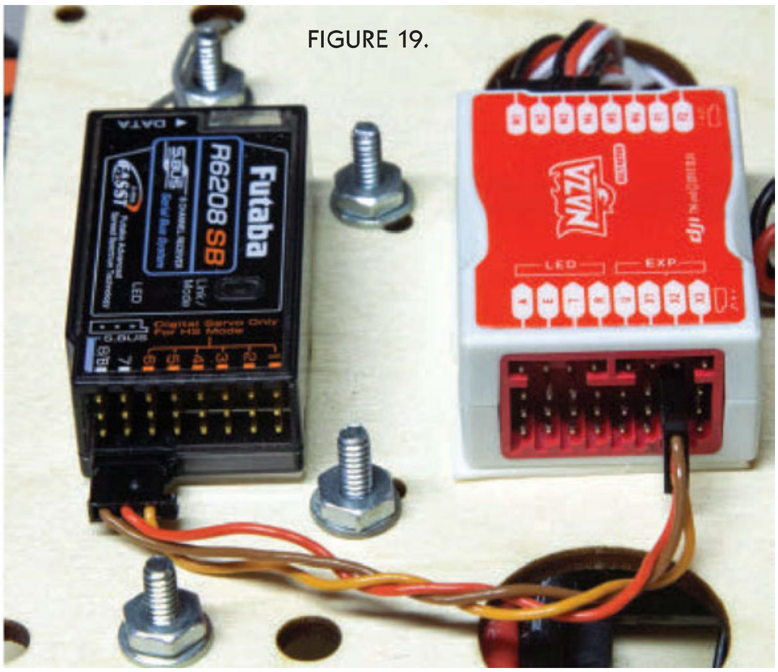
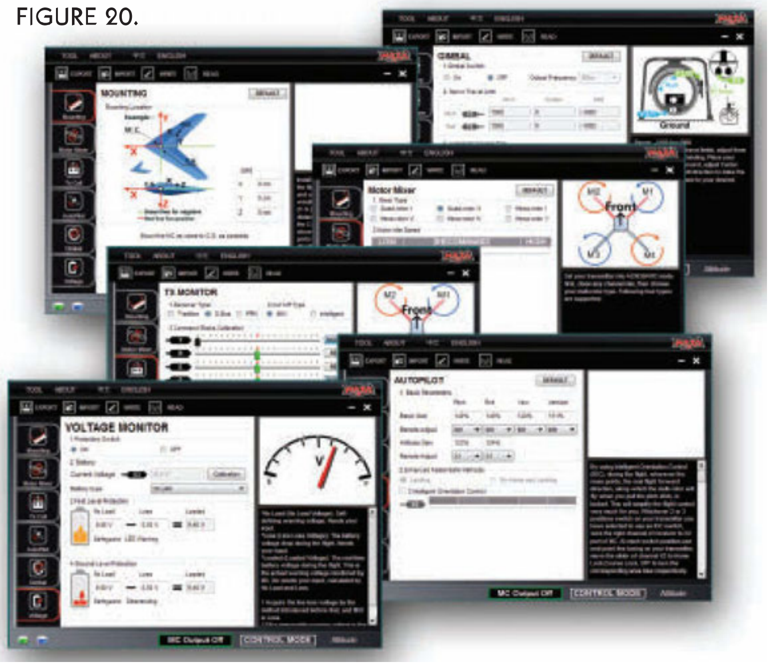


FIGURE 19.

FIGURE 20.



VU Connections

The VU has three connections. The first — which you have already connected — is the battery. The second is the three-pin connector that looks much like a servo connector. This one connects to the NAZA pins labeled X3. The last is a four-pin connector that plugs into the set of four pins labeled LED. Route the cables down through the hole near the VU and then back up through the hole in the rear near the NAZA as shown in **Figure 18**.

Futaba S.BUS

As an option, both the NAZA and Futaba support a special radio protocol called S.BUS. This allows you to use a single cable connected between the Futaba S.BUS connector and X2 on the NAZA. **Figure 19** shows the connection. This makes wiring much less complicated.

NAZA Configuration

Before you attach your propellers, you will need to configure your NAZA. This is done using the DJI NAZA Assistant software. The software has six screens which are shown in **Figure 20**. Through these screens, you set the base configuration of your craft, calibrate your radio, and set the VU voltage levels.

The NAZA assistant software, as well as the USB driver for the NAZA can be downloaded at **www.dji-innovations.com/products/naza-multi-rotor/downloads**.

I will be going over each NAZA configuration page in detail on my website at **www.kronos robotics.com/multirotor/kronosflyer**.

Please be sure to finish the configuration before installing your props.

Prop Installation

After you calibrate and program your ESCs, set your motor directions, and install and configure your flight controller, you can install your props. You leave the props for last because you don't want your craft flying away as you make adjustments.

Before you attach the props, you must balance them. There are a couple of ways to do this, but I have found the most reliable way is to use sand paper to sand the heavy side of the prop. I will add some more detailed instructions for balancing your props on my website. You can also search the Web. There are videos that will take you through the process.

Each prop comes with a set of shaft adapter inserts so that they fit perfectly on various motor shafts. You want to use the 5 mm insert for both the NTM and SK3 motors. Insert the adapter into the back side of the prop. Next, insert the prop onto the

motor shaft, back side first. Note that the back side of the prop is the side that has no writing on it.

Add the prop washer, then the nut as shown in **Figure 21**. Note that the NTM motors have a spinner instead of a nut. Tighten the nut/spinner.

Not all your props are the same. Two of them are left turning props and two are right turning props. The props marked with the R should be installed on motors 2 and 4. If the prop nomenclature does not have an R, it is assumed to be left turning, and should be installed on motors 1 and 3. Refer again to **Figure 8** for the motor numbers.

FIGURE 21.

Conclusion

So, you have gone through four months of gathering parts and assembly. You want to actually get your Kronos Flyer soaring. This is difficult to cover in an article, so I uploaded a video to both my website and at the article link.

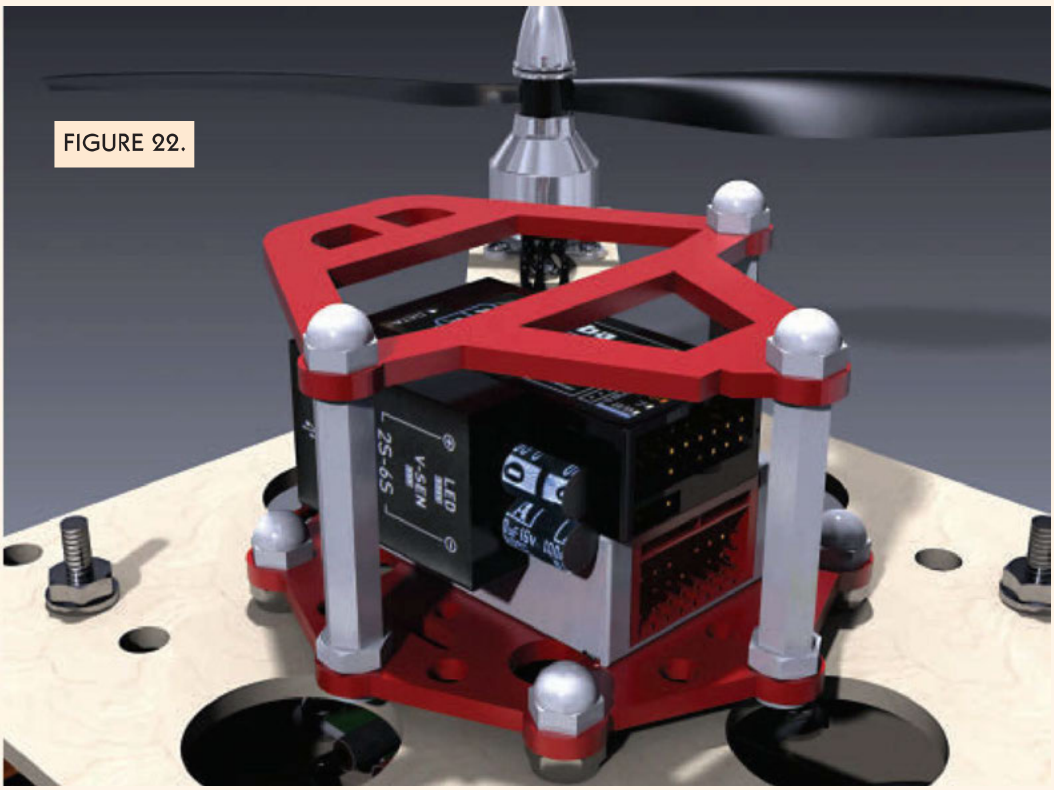


FIGURE 22.

UNM

I own several craft that I have built, and while the cost of the frame components, ESCs, motors, and hardware can add up, they are nowhere near the cost of a NAZA flight controller -- especially if you have purchased the GPS upgrade. For this reason, I created the Universal NAZA Mount (UNM).

Your NAZA, VU, receiver, and (optional) GPS mount onto the UNM platform. The UNM platform mounts to your components platform. The only external connections are the VU power and the four ESC connections. This allows you to move the NAZA and radio to another craft with little effort. It only takes a couple of minutes.

Shown in **Figure 22**, the UNM is held in place by four acorn nuts that attach to the four inner screws on your component platform.

More information on the UNM can be found on my web site at **www.kronosrobotics.com**.

Next Month

Next month, I will close out this series. I will go over upgrade options and some of my future plans for the Kronos Flyer.

Feel free to ask me questions directly on the *SERVO Magazine* forums or at the article link. **SV**

Santa SERVO Surprises Upstate New York

by Dave Graham

The city of Johnstown in upstate New York roared to life on Friday night, November 16, 2012 as 70 groups from Fulton, Montgomery, and Saratoga counties converged on the town to display their floats and to march in the annual holiday parade. Leading the charge this year was a float designed as a collaborative effort between the Electrical Technology (Electech) Club and the Radiology Technology Program of Fulton Montgomery Community College (FMCC).

Electech Club President Heath Hill took the lead and got to work early planning and building the float (**Figure 1**) with a student-designed computer controlled light show. The light show consisted of 25 strands of multi-colored lights controlled by a Parallax BASIC Stamp driving a relay control board (**Figure 2**). Hill used a Honda 2KW generator to power the float. The Radiology program added their special touch to the show by following the float with a skeleton in a wheelchair.

Electech Club members are all avid readers of *SERVO Magazine* (**Figure 3**), and as such, thought it would be a nice twist to hand out copies of the magazine to the parade watchers rather than just the usual candy. It would be their way of injecting technology into the event and generating interest in the various FMCC technology programs with the spectators. Not sure how to ask for the magazines, Electech President Hill wrote a letter to Santa SERVO and asked for some help. Santa SERVO didn't disappoint, and delivered 40 pounds of *SERVO Magazines* to the Electech Club float for students to distribute during the parade. Hill was amazed while handing out magazines during the parade (**Figure 4**) that the

FIGURE 1. Electech Club float under construction (note the boxes of *SERVOs* at the lower left).

FIGURE 2. Float light show controller.

FIGURE 3. Electech Club members reading *SERVO Magazine*.

spectators preferred the robotics magazines over the candy! The Electech Club would like to thank the following individuals/organizations for their contributions to the Electech Club float (**Figure 5**):

President: Heath Hill
Vice President: Angelina Brooks
Secretary/Treasurer: Mary Cedillo
Programmer: Stephan Sampson
Network Technician: Kyle Bonfey
Float Design Coordinator: Nicholas Stewart
Student Government Representative: Chris Renda
Float Trailer and Hay: Frank Ryan Jr. and Lee Hollandbeck
Honda 2KW Generator: Ken Kight
The Radiology Technology Program
SERVO Magazine: Santa SERVO **SV**

FIGURE 4. Electech Club President Heath Hill working the crowd during the parade.

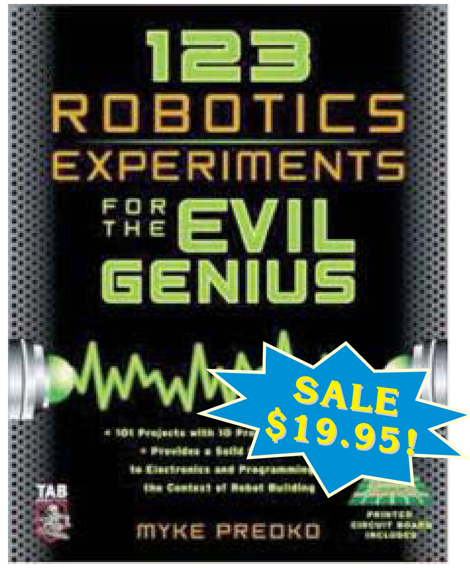
FIGURE 5. Electech Club float and supporters.

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS

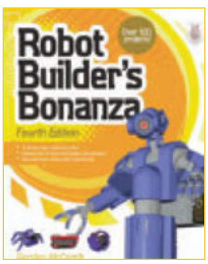
02.2013 FEATURED



ROBOTICS

Robot Builder's Bonanza, Fourth Edition

by Gordon McComb

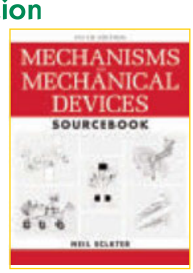


Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.

$29.95*

Mechanisms and Mechanical Devices Sourcebook 5th Edition

by Neil Sclater

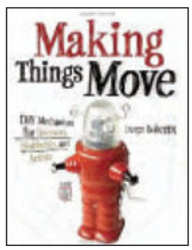


Fully revised throughout, this abundantly illustrated reference describes proven mechanisms and mechanical devices. Each illustration represents a design concept that can easily be recycled for use in new or modified mechanical, electromechanical, or mechatronic products. Tutorials on the basics of mechanisms and motion control systems introduce you to those subjects or act as a refresher.

Reg $89.95 Sale Price $79.95

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts



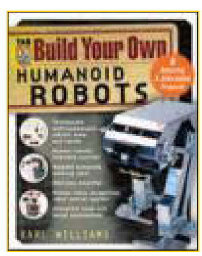
In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

$29.95*

Build Your Own Humanoid Robots

by Karl Williams

GREAT 'DROIDS, INDEED!

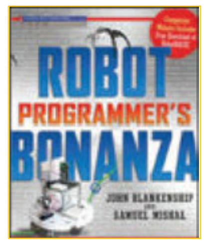


This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot.

$24.95*

Robot Programmer's Bonanza

by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!

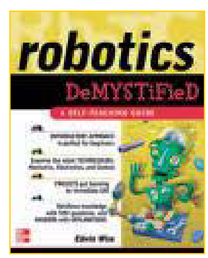
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

$29.95

Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

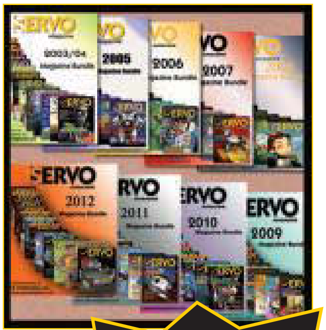


Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles



Save $10.00
Only $57.95!

Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, 09, 10, 11 and 2012.

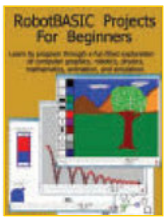
Any bot builders out there?
Get cool robotics stuff from my store!

Call me at my order desk!

Visit my online store @ www.servomagazine.com

RobotBASIC Projects For Beginners

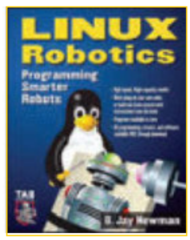
by John Blankenship, Samuel Mishal



If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.
Price $14.95

Linux Robotics

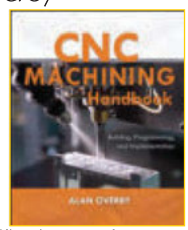
by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.
$38.95

CNC Machining Handbook: Building, Programming, and Implementation

by Alan Overby

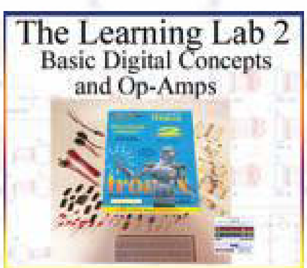


The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource. **$34.95**

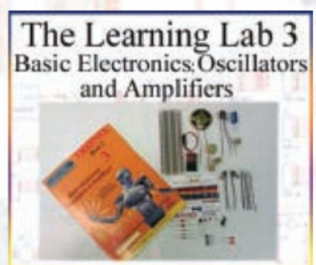
FOR BEGINNER BOT BUILDERS

New Lower Price!

$59.95



$49.95



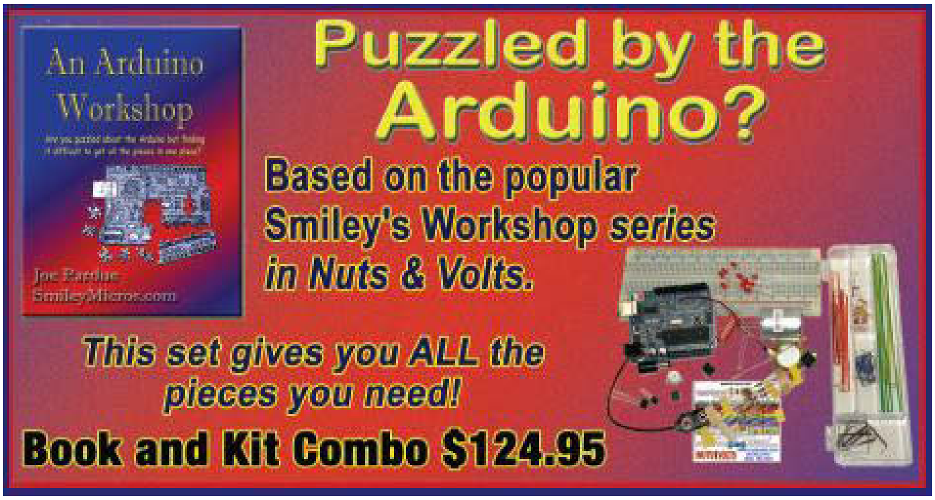
$39.95

The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.

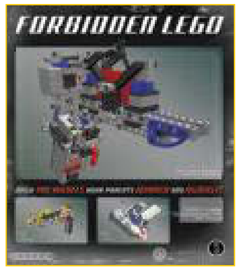
Or order online www.servomagazine.com

SPECIAL OFFERS



Forbidden LEGO

by Ulrik Pilegaard / Mike Dooley

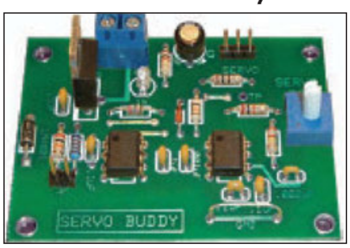


Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

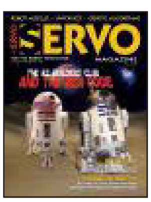
Reg $24.95 Sale Price $19.95

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.



For more information, please check out the May 2008 issue or go to the SERVO webstore.

Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

3D LED Cube Kit

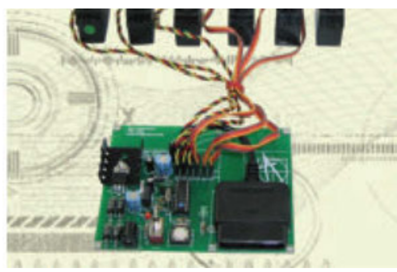
From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of *Nuts & Volts Magazine*.

This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes.
Colors available: Green, Red, Yellow & Blue.
Jig and plastic cases also available.

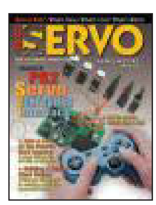
Subscriber's Price **$57.95**

Non-Subscriber's Price **$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.



For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price **$79.95**

Non-Subscriber's Price **$84.95**

Part 2

Using Digital Sensors With VEX

By Daniel Ramirez

Discuss this article in the *SERVO Magazine* forums at **http://forum.servomagazine.com**.

The 2007 movie, *The Golden Compass*, is an exciting fantasy-adventure film that was based on *Northern Lights* — the first novel in Philip Pullman's trilogy, *His Dark Materials*. It stars Nicole Kidman, Sam Elliot, Eva Green, Dakota Blue Richards, and Daniel Craig. The story depicts the adventures of Lyra Belacqua — an orphan living in a parallel universe on a world that looks much like our own. In Lyra's world, a ruling power called the Magisterium is conspiring to end tolerance and free inquiry. Poor, orphan, and Gyptian children are disappearing at the hands of a group the children call the Gobblers. Lyra discovers that Mrs. Coulter is running the Gobblers. Rescued by the Gyptians, Lyra joins them on a trip to the far north in search of the missing children.

The primary device used by Lyra in this adventure film is a golden compass that always points to the truth. It gave me the idea to build this DIY Electro-Mechanical (EM) compass which I'll finish describing in this article. While this compass won't point to the truth, it will point you in the right direction.

In Part 1, I showed you how to build the gearbox with the VEX motor that is part of the EM compass' motion subsystem. This time — in addition to showing you how to finish the EM compass — we will provide you with the I^2C firmware necessary to make the EM compass work, and even more information on other kinds of I^2C devices that can be connected using these drivers. You'll also learn how to integrate electronic compasses into your own VEX robotics applications using the information provided here.

ELECTRO-MECHANICAL COMPASS ASSEMBLY

To finish this compass, you will need to refer to Part 1, so it's a good idea to have the January 2013 issue handy. Recall that construction details for the gearbox were covered in Part 1. A trip to your local arts and crafts and office supply store is definitely in order if you want to customize the compass using the materials shown in the updated bill of materials in **Table 1**. The compass can be made as large as is practical using the VEX construction set and gearbox. I chose a 14" diameter and 6" high cylindrical hat box that I bought at a local store, using the lid for the compass and the box for the enclosure.

For safety's sake, please make sure to wear goggles and gloves when hot gluing and drilling holes necessary for the following steps.

The first step is to locate the center of the hat box lid

TABLE 1. The updated bill of materials.

QTY	DESCRIPTION	SOURCE
1	VEX microcontroller	Innovation First, Inc. (IFI) **www.vexrobotics.com**
1	9.6 or 7.2 volt battery	IFI
1	30-gauge wire-wrap or stranded wire	RadioShack **www.radioshack.com**
1	VEX bumper switch*	IFI
1	HMC6352 electronic compass	SparkFun **www.sparkfun.com**
1	12-tooth gears: 0.5" (12.7 mm) pitch dia.	IFI
1	36-tooth gears: 1.5" (38.1 mm) pitch dia.	IFI
1	60-tooth gears: 2.5" (63.5 mm) pitch dia.	IFI
1	84-tooth gears: 3.5" (88.9 mm) pitch dia.	IFI
1	VEX quadrature encoder*	IFI
4	Square metal shafts	IFI
1	VEX three-wire motor	IFI
1	14"diameter x 6" H cylindrical hat box	Local arts and crafts store
1	Gold or silver foil paper	Local arts and crafts store
1	Circular protractor (0-360 degrees)	Local arts and crafts store
4	2" aluminum standoffs	IFI

*Optional

and carefully drill a 1/4" hole in both the center of the compass base and the compass rose pattern in order to provide room for the compass motor axle, and also to pass through four thin wires around 20" in length (in order to leave plenty of slack). Use 30-gauge stranded wire since these wires will be twisting around when the compass pointer moves.

At this point, you can route the thin wires leading from the electronic compass (available from SparkFun) through the 1/4" hole in the center of the compass base to the VEX microcontroller underneath. The EM compass base should now have a (preferably laminated) compass rose pattern glued onto the inside of the hat box lid with the 1/4" hole drilled in the center.

The second step is to carefully mount the VEX gearbox underneath the compass base, which is supported with four aluminum standoffs that are each 2" long. Hot glue them to the bottom of the compass enclosure. The 3" axle leading from the largest gear should be inserted into the 1/4" hole and carefully centered. The compass pointer is then attached to the other end of the axle. Use four aluminum standoffs to mount the gearbox underneath, then mark their positions and hot glue them to the top of the box lid.

The completed gear — which is necessary to drive the compass pointer using a standard three-wire motor — is shown in **Figure 1**. Gear reduction allows you to control the compass movements using very small motion increments. Note the optional VEX quadrature encoder that can be used for calibrating the compass and for setting the compass headings. You can also see a 9.6 volt battery fastened to the bottom of the compass base.

Finally, paint the pointer using model enamel paint. I used red and black to indicate the north and south directions, respectively, as is the case with most magnetic compasses. However, any two colors can be used. Take a look at **Figure 2A**.

I enlarged a standard protractor using a copier so that it would surround the 14" diameter base. This enables you to take readings in degrees, and to calibrate the compass and adjust for magnetic declination.

You can "steampunk" the EM compass by gluing on a cool compass rose pattern (like the one

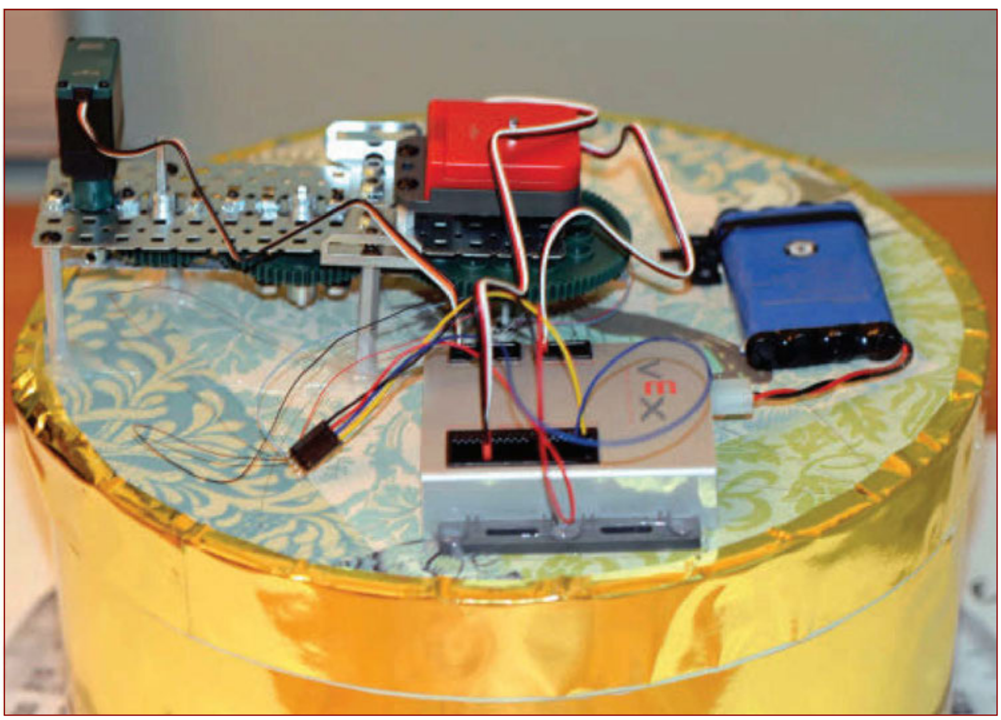
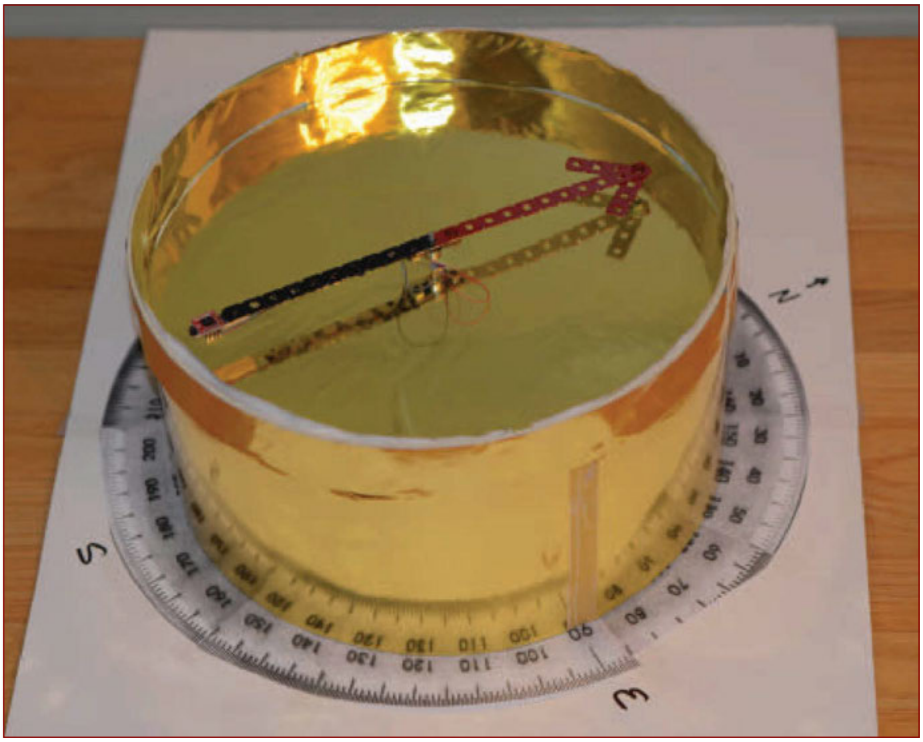


FIGURE 1. The completed gearbox that is attached to the compass base. This is necessary to drive the compass pointer using a standard three-wire motor. Gear reduction allows you to control the compass movements using very small motion increments.

FIGURE 2A. Note the color scheme and addition of the protractor markings.



provided in Part 1) and using a custom color scheme. As you can see, I chose gold foil paper for mine in keeping with the movie. Glue the compass rose pattern you have selected to the bottom of the hat box lid. In addition, you can line the inner and outer lid rim to form a decorative golden ring.

The completed compass showing the four cardinal compass points (N, E, S, W) along with the protractor scale indicating angles between 0 and 360 degrees (where true north is 0 degrees) is shown in **Figure 2B**. As you can see, this 14" diameter compass is much larger than the standard magnetic compass shown next to it.

Although I have not done so yet, the top of the compass can be covered with a clear lid made from a circular cutout of plastic or glass for both weatherproofing and to keep hands away from the pointer during operation.

ELECTRICAL CONNECTIONS

Refer back to the schematic in Part 1 to make the final electrical connections using the four thin wires that you routed through the 1/4" hole in the center of the compass base to the VEX microcontroller underneath for SCL, SDA, +5V, and GND. The optional pushbutton switch can be used while calibrating the compass with the quadrature optical encoder.

The final step is to clean up the assembly by removing excess glue and covering up any screws that are visible. This completes the construction phase of the EM compass.

I^{2}C FIRMWARE FOR THE ELECTRO-MECHANICAL COMPASS

In Part 1, we described the I^{2}C serial two-wire protocol used to communicate with digital sensors such our compass, leaving the firmware details for this part. So, let's dig into the firmware needed to make the microcontroller work with our compass.

You will find that this firmware works with other kinds of I^{2}C devices by tailoring the main driver and customizing it for the device to be connected. In order to do this, you need to have the latest datasheet for the device in question. In this case, the datasheet for our compass is

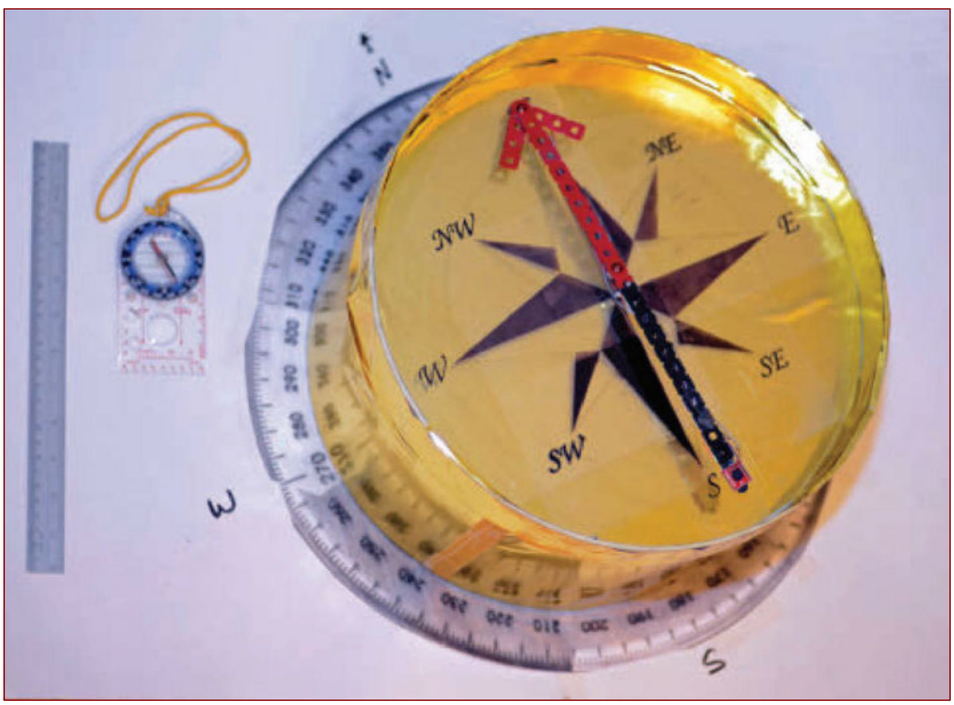
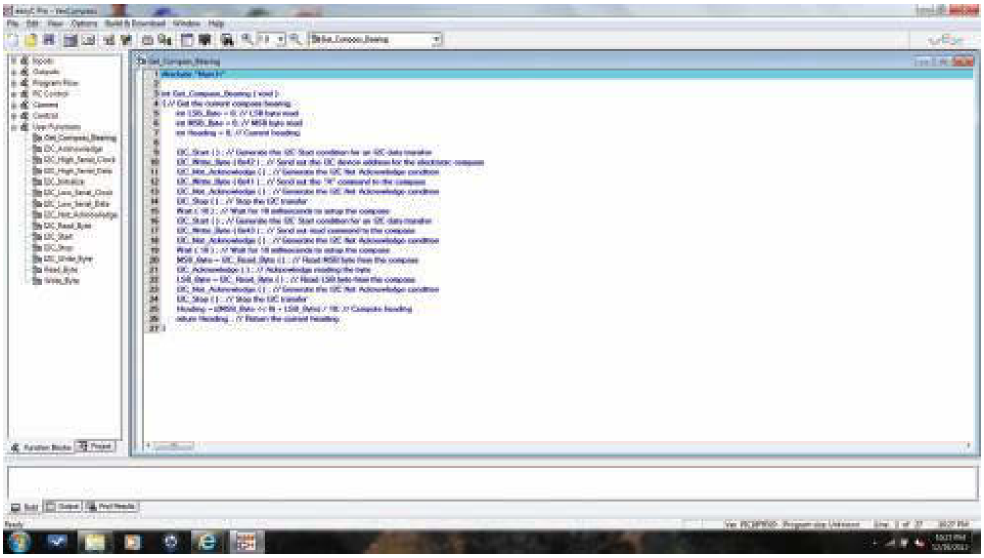


FIGURE 2B. The completed EM compass with the pointer and the four cardinal compass points, along with a protractor scale indicating angles between 0 and 360 degrees.



LISTING 1. The EM compass is tested for basic operation using the *Get_Compass_Bearing* routine. If all is working correctly with the I^{2}C firmware, you should see readings that are continuously changing to values between 0 and 360 degrees.

located at **www.sparkfun.com**. We will use it to determine what I^{2}C command bytes need to be sent to the compass module, what responses are required to initialize it, and then actually get the compass readings. This is shown in **Listing 2**, which demonstrates how to initialize the compass module by sending the necessary I^{2}C commands that are bit banged (simulated) using the compass application. **Listing 3** shows you how to bit bang an I^{2}C Start condition which is used to begin a transfer.

Last month, we also covered the I^{2}C conditions which make up this routine using the I^{2}C serial two-wire protocol. Remember, every I^{2}C device has its own unique address and we described the format which included a seven-bit or 14-bit address. The commands are sent to the connected I^{2}C device and are specified by the address (using the drivers provided with the bit banged routines). In this case, we use the address of the SparkFun I^{2}C module which is 0x42. **Listing 3** provides a more detailed example with the *I2C_Start* routine. The remaining routines include *I2C_Stop*, *I2C_Read*, *I2C_Write*, etc., so use similar code to that shown here.

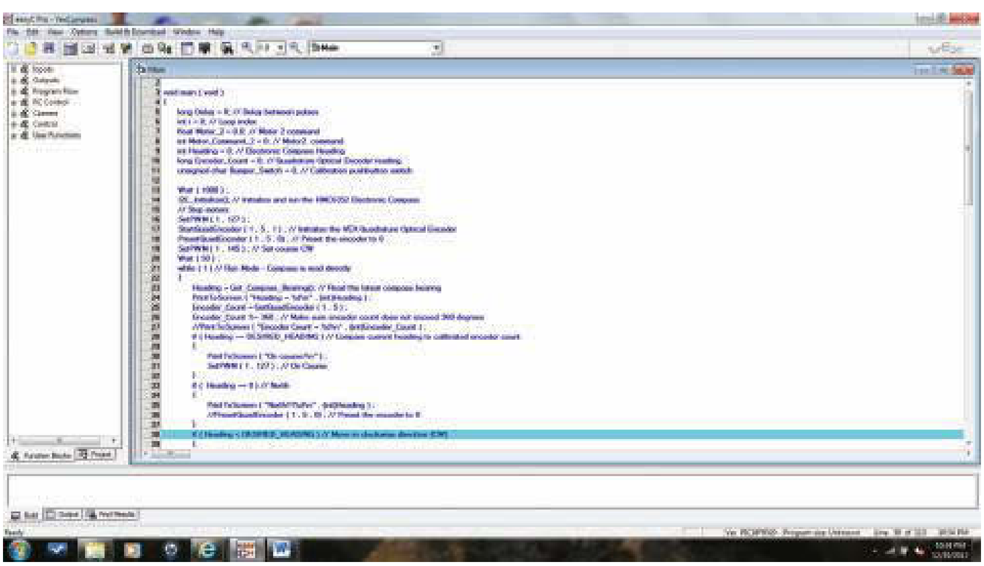
Recall that these low level drivers can be used with other I^{2}C devices although the communications speed will be slower than it otherwise would be if you were using direct I^{2}C hardware support such as that on a VEX Cortex or VEXpro (ARM9). Source code is available at the article link.

TESTING THE COMPASS

At this point, the compass is now ready to be calibrated and tested. This is done by first downloading the main compass application to the microcontroller using the Easy C Pro bootloader. For testing purposes, detach the compass pointer so that the wires don't get tangled. Also, make sure to keep the pointer level (use a carpenter's level if necessary).

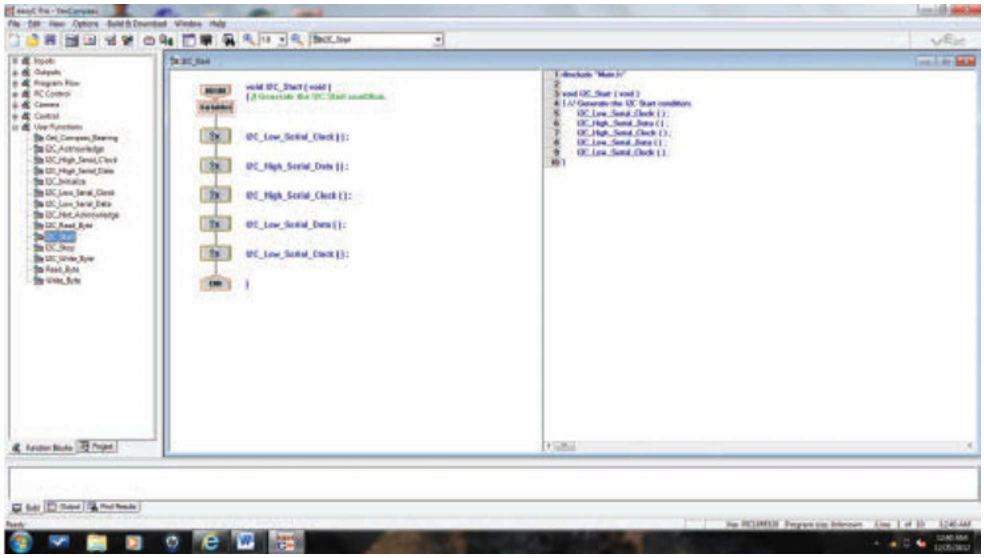
We test for basic operation using the *Get_Compass_Bearing* function shown in **Listing 1**. Notice how the waveforms are generated for the necessary conditions, including the *I2C_Start*, *I2C_Write_Byte*, etc., which are required by the I^{2}C protocol. The complete list of bit banged primitives is shown on the far left. A fragment of the main application code that calls the *Get_Compass_Bearing* function to do this test is shown in **Listing 2**.

The compass angles returned from the application are shown in **Figure 3** which is a screen capture of the data generated by the test program. If all is working correctly with the firmware, you should see compass readings that are continuously changing to values between 0 and 360 degrees as you slowly and carefully move the pointer around (remember to keep



LISTING 2. Performs a proportional control loop to keep the compass pointer angles synchronized with the readings. This code can also be used to help robots stay on course.

LISTING 3. An example of how to bit bang an I²C Start condition which is used to begin a byte transfer. Recall that we covered these I²C conditions which make up the serial two-wire protocol in Part 1.



the pointer level). The key is that the readings should be decimal values between 0 and 360, and should change depending on the compass orientation. The pointer will slowly keep trying to point to zero degrees north and will move accordingly using the proportional control algorithm.

If the value gets stuck, there may be a wiring problem or the electronic compass module may not be working. If the wires start getting tangled, switch the microcontroller off to avoid damage to the compass. Notice that if you align the optional quadrature encoder to start at 0 (north), it will also return the compass direction in degrees.

This compass application insures that all electro-mechanical components (including the VEX gearbox) are functioning correctly, and are continuously reading and displaying the compass bearings.

Please note: Before powering up the VEX microcontroller, make sure that there are no objects (or fingers or hands) inside the compass rim. Also, as mentioned previously, be sure to wear safety eyeglasses and gloves. The compass pointer could move inadvertently causing injury. Please keep a safe distance from the pointer while it's in use.

COMPASS CALIBRATION

Before starting the calibration process, make sure the compass is level. It is calibrated by aligning the pointer to magnetic north and adjusting for declination so that it points to true north. You will need to modify the application by adding the magnetic declination for your specific geographic location; you can do this using the online NOAA calculators described below. In addition, you can align the quadrature optical encoder so that it starts counting at 0 degrees (north). The optional bumper switch can be used for this purpose.

Delays between motor commands in the firmware may need to be adjusted slightly to move the compass pointer correctly, depending on the pointer's mass and rotational friction. Also take into account the electrical wires since the motor may not have enough torque to control the pointer. A shorter delay decreases the torque, while a longer delay increases it.

Remember to keep the compass as far away from magnetic metals as possible, so they don't interfere with the readings.

In order to keep the compass pointing to true north, the microcontroller performs a simple proportional control loop written in Easy C Pro (shown back in **Listing 2**) to keep the pointer angles synchronized with the readings. This code can also be used to help robots stay on course by

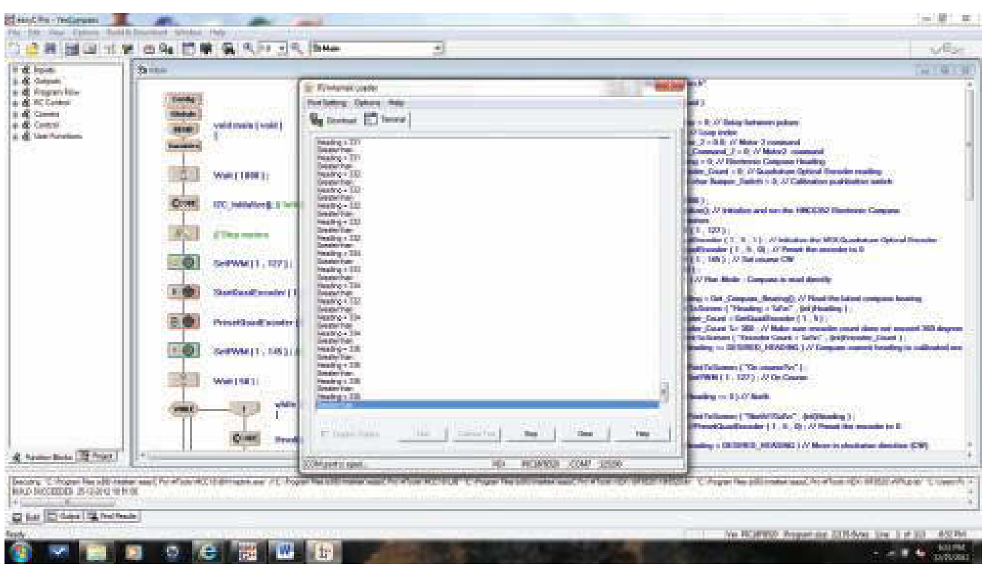


FIGURE 3. Screen capture of the data generated by this application. Note that we will probably get different compass readings, depending on the initial compass orientation. The key is that the readings should be decimal values between 0 and 360.

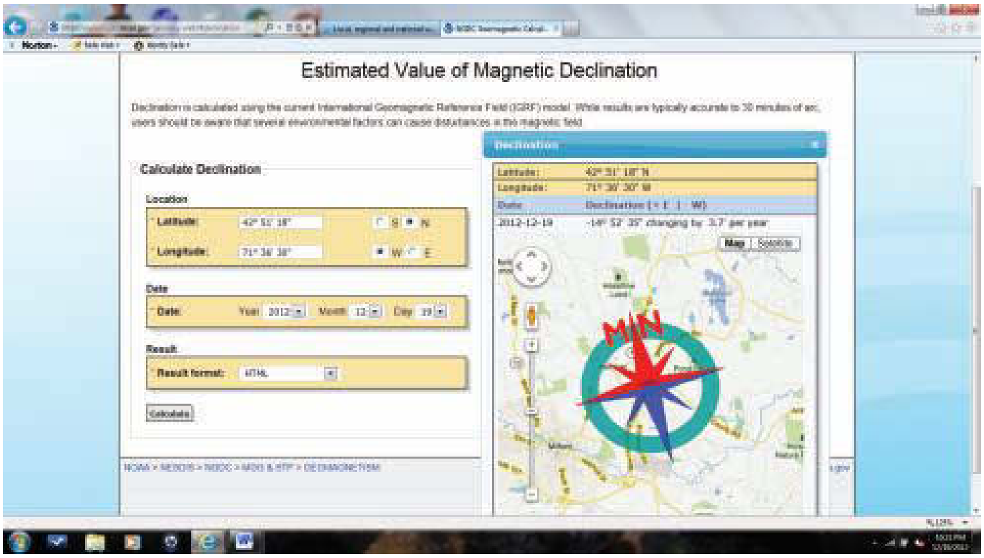


FIGURE 4. You will need to adjust the compass readings with the magnetic declination for your particular location so that they will point to true north.

maintaining a specific heading set in the *DESIRED_HEADING* global constant. It also takes advantage of the quadrature optical encoder. Finally, in the bit banged *I2C_Start* routine in **Listing 3**, notice how the waveforms for this Start condition are synthesized using low level clock and data bit toggling routines.

As mentioned, using the proportional control loop can help a robot navigate a specified course. This is done by having the microcontroller issue small motor commands that are geared down using the VEX gearbox, which generates tiny motion increments.

Your compass should now be fully operational.

I²C HARDWARE SUPPORT FOR VEX CORTEX OR VEXPRO MICROCONTROLLERS

You can program a VEX Cortex or even a VEXpro microcontroller to use I²C hardware ports by utilizing the routines provided in EasyC Pro or Robot C (or even QWERK C++ for the VEXpro, once I²C hardware drivers become available from the respective compiler vendors). Until then, you need to bit bang the I²C routines in a similar fashion to how we did them here for the microcontroller. In either case, I don't believe the main application would need to change much (although I have not tried it).

I²C support for the Cortex and VEXpro is provided directly with a hardware port that is used to read the new integrated quadrature encoders that work with the new VEX 393 motors. They use the I²C hardware to generate the necessary signals, and to read the data using Easy C or Robot C for the Cortex.

One problem with using I²C on these platforms is that the support is not part of the EasyC or Robot C language yet. When it does become available, you should be able to run the main I²C application for the compass with minimal changes; just substitute the names of the bit banged routines for the low level I2C routines provided by EasyC Pro or Robot C.

USING THE EM COMPASS

So, how do we get true bearings? By adjusting the EM compass readings with the declination from true north or magnetic north based on your geographic location. An easy way to do this is to use the magnetic declination calculator provided by NOAA at **www.ngdc.noaa.gov/geomag/declination.shtml** as shown in **Figure 4**.

This is where you can make use of the optional quadrature encoder to enter the declination angle instead of having to modify the application. For this adjustment, we use the moving ring on the compass to compensate for declination from true north by adding or subtracting the declination for our geographic location (latitude and longitude). While these abnormalities are mostly caused by the earth's molten, they can be adjusted using the on-line declination compensation calculator. You simply enter your location (or zip code for the USA) and get the declination value. **Remember when using the declination values that east declination is positive; west is negative.** This value is then added to the bearing returned from the *Get_Compass_Bearing* function in the code.

USING OTHER KINDS OF I²C DEVICeS WITH VEX

Now that we know how to make the VEX 0.5 microcontroller work with our electronic compass, we can use the firmware to work with other kinds of I²C sensors and devices such as math coprocessors, serial EEPROMs, and even LCD displays.

This can be done by tailoring the main I²C application and customizing it for the specific device in question using its datasheet (as discussed earlier). To do this, we look up the necessary I²C command bytes on the respective datasheet to initialize the device and get it started taking sensor readings, or we can send it command bytes and receive back data bytes.

Using some of these sensors, we could easily construct a weather station, a home security system, or an advanced robot. Some of these new I²C sensors include:

- XYZ accelerometers
- Gyros
- Temperature sensors
- Pressure sensors
- Humidity sensors
- Ultrasonic rangers

IMPROVEMENTS

There are some improvements that can be made to the compass which I did not get around to doing. First, the accuracy can be improved by collecting multiple readings, say 20, and then averaging them. This is known as signal conditioning or filtering, and is usually done when reading noisy sensors. Another improvement is to install a power switch in back of the compass to turn the power to the microcontroller on/off without having to remove the compass from its base. You can also make better use of the quadrature optical encoder and bumper switch to enter headings and magnetic declination offsets.

SUMMARY

In this installment, we completed the construction details of our EM compass using VEX components, and also showed you how it could be sort of steampunked to customize its appearance using materials obtained from local stores.

I also described how to use the quadrature optical encoder to set the magnetic declination input or to set a navigational course for your robot.

The I^2C drivers required to access the electronic compass module were given. In addition, I showed you how to test and calibrate the compass so that you could use it to obtain actual compass bearing readings that were corrected for declination in order to point to the "true" bearing. Finally, we briefly discussed how other kinds of I^2C devices including coprocessors, LCD displays, memory ICs, etc., could use the information provided here.

The Microchip 24LC1024 serial I^2C EEPROM, for example, is a very useful device which can be connected to the VEX microcontroller using the firmware provided here. It can be used as a memory expansion upgrade to store large lookup tables and calibration data, or even to store data collected from various sensors. If there is enough interest in this, I will include it in a future article.

Next time, we will cover another high speed protocol known as the SPI interface. Until VEX time. **SV**

Then and NOW

Robot Telepresence and Personal Assistants

by Tom Carroll

Back in 2006, I wrote a few columns about how robots lived with humans and how they cared for people. No matter how many AUV robots we might see patrolling our roads and skies, the most interesting robots to most of us are those that work amongst us. Large corporations, universities, and 'think tanks' like Willow Garage are furiously developing autonomous robots for the home environment. In the seven years since I researched those articles, there have been some amazing machines developed that have taken their place beside us humans. Every lab and company in this field is striving to design and build that next perfect home companion robot.

Bill Gates will probably never be on everyone's 'favorites' list, but his comments and quotes always generate interest within the technical community. I personally feel that he is right far more than he is wrong — especially concerning robotics.

A quote of Gates was included in Tom Friedman's 2005 book, *The World is Flat*: "There are one hundred universities making contributions to robotics. And each one is saying that the other is doing it all wrong, or my piece actually fits together with theirs."

I remember this comment from back then, and it has intrigued me ever since. Intense competition is always present within companies, but it seems even more so in robot development.

Just as television technology has progressed from crude black and white CRT sets, to color, to digital, and now to side-lit LED LCD flat screens, robots are also making major leaps in technology. Personal and service robot research and development certainly has been fierce, but the goals of robot developers are far more than just bringing forth that one killer app.

Robotics is much too complex to have one particular invention or design become the key that revolutionizes the field. It will take universities and other researchers working together with industry to deliver a true personal home companion robot, with each contributing their own piece of the puzzle for the good of the industry.

I would like to highlight some of the latest happenings in personal and home service robot development.

Telepresence Robots

I would like to begin our discussion of robots working around people with a look at some of the latest telepresence robots. Telepresence or (as some like to say) remote presence has actually been with us a long time as a camera atop a computer using software such as Skype. Teleconferencing has been used for commercial applications even longer. The addition of a mobile base and a wireless interconnection should make this type of two-way TV robot

FIGURE 1. InTouch Health RP-7 robot.

among the most simple and least expensive to design and build, but the high costs of some of the latest telepresence robots might make a prospective buyer wonder.

FIGURE 2. AnyBots QB telepresence robots.

The appeal to potential buyers is to have a mobile remote two-way camera and audio system that they can personally maneuver within a group of people to give the illusion of their presence within that group. **Figure 1** shows an RP-7 remote presence robot designed by InTouch Health back in 2007 that gives a remotely located doctor the ability to confer with a patient. iRobot has since joined with InTouch and the advanced RP-VITA was developed in conjunction with the two companies. I will discuss this robot in detail later.

Telepresence Robot Design

The design of a telepresence or remote presence robot is the basis of most personal robots. As mentioned, it is a simple and straightforward design with only two drive motors to deal with. Many of today's robots for the home have some sort of video camera or image recognition camera in the 'head' or upper part of the robot. Many have included an LCD display to present information for the operation of the robot, or to show an image of a person using the robot by remote control. The camera and display work best if they are at least four feet above the floor, so this height requirement calls for a tall robot.

Today's cameras are small, but displays can easily be a few pounds in weight, depending on their size. The AnyBots QB telepresence robots shown in a long line in **Figure 2** use a smaller LCD display and a rubber bumper around their heads.

Even if you have a lightweight display and associated mounting (or camera or possibly a speaker) placed up high, it can result in a tipsy robot. A larger LCD screen works and looks much better than a small smartphone-sized display in my opinion, though some of the latest robots do sport a tablet or smartphone at the top. A pair of beefy drive motors and a lead acid battery pack in the base can compensate for most (but not all) of the mass in the head. A quick start, stop, or turn can easily cause the head's small amount of mass to overcome the low center of mass, and tip the robot over.

Enlarging the wheelbase can help, but having a built-in speed control that creates a ramped-up or ramped-down drive signal to the motors adds to the stability. **Figure 3** shows a very nice do-it-yourself telepresence robot (from the brantlew blog site) that uses an iRobot Create base and a laptop computer. I can only hope that the laptop is expendable or extremely tough.

The iRobot Create only has a five pound load limit and spring-loaded drive wheels that make it a bit wiggly for some robot designs. It can be a great experimenter's robot platform, but a Parallax Madeusa, for example, or a similar larger base is more stable and sturdier for tall robot designs.

Virtually all of today's telepresence robots use differential drive with two driven wheels. Most use a fore and aft caster wheel for stabilization, but some use two-wheel dynamic balancing, Segway style. This works fine until a computer or power glitch causes the robot to crash on its face. (QB's rubber bumper ring would sure come in handy.)

A few designs with two-wheel balancing have stabilizing feet that quickly descend in an emergency or while the robot is at its charging station. Differential drive (or tank

FIGURE 3. DIY telepresence robot from the brantlew blog site.

style) is great for a telepresence robot because it allows the robot to turn on its own axis — much like a panning TV camera.

Robo Dynamics — From Telepresence to Personal Robot

Most of today's designs evolved from earlier prototypes. **Figure 4** shows the Robo Dynamics TiLR that debuted back in 2008. TiLR stands for Telepresence Internet-connected Low cost Robot. At a cost of $10,000, it is not low enough in cost for the average homeowner, but it was one of the first available for the corporate consumer.

TiLR has some great features that are still not available on most of today's telepresence robots. The pan and tilt zoom camera is one of its best assets. It allows the operator to place the robot in one location within a conference room and pan the camera to each person within the room who happens to be speaking, without having to rotate the robot's body. A 26x optical zoom feature allows the remote operator to get up close and personal with the speaker's face. All these control features can be accomplished with just a laptop computer and a web camera from anywhere in the world.

The camera can also be tilted up

FIGURE 4. Robo Dynamics TiLR 2008.

and down. For general viewing within a room full of people, the camera points straight ahead, whereas driving the robot around a room or through doorways is best done with the camera at a 45 degree down angle. For tight maneuvering among obstacles and/or cables on a floor, the camera can be pointed straight down. The large LCD screen in the head makes viewing of the remote person a lot easier than with the smaller smartphone screens of later designs.

The software for the laptop that's used to remotely control the TiLR allows for the programming of individual people within a room. The remote operator can then click on a pre-programmed designated square 'bookmark' on the laptop's screen and the camera will immediately go to that particular person's face, without the need for the operator to search by panning the camera. The TiLR is heavy but very well made, and, as mentioned, is a bit too expensive for wide use outside of major corporations.

In May 2011, Robo Dynamics announced Luna, which is considered a personal robot. Shown in **Figure 5**, Luna is designed for people to use at home. It is one of the most aesthetically pleasing robots that I have seen. Its sleek lines and teardrop body shape allow it to easily live within one's home; yet, there is no mistake that it is a robot. The Santa Monica, CA based company designed the robot to be affordable at an introductory price of $3,000, but hope to drop the price further to around $1,000.

Luna has a dual core Intel Atom processor running Linux-based LunaOS, and uses an eight inch LCD touch screen. It can be programmed for basic functions and has positional feedback from the two wheel drive motors. The vision system is via an 8 MP camera with digital zoom, and it hears via three onboard microphones. Standing 5'2" with a 22" diameter base, it weighs a svelte 65 pounds. It has a 26 Ah sealed lead acid battery and can operate four to eight hours with the same charging time. The 'arms' don't move but they can be manually positioned to carry a tray or something similar. I can see a robot developer buying one of these robots and building some very functional arms and grippers to make this a truly amazing personal robot.

FIGURE 5. Robo Dynamics Luna.

Robo Dynamics CEO Fred Nikghar wants to make Luna the robot equivalent of the smartphone — available to the average consumer. He sees his less complex LunaOS as easier to use when compared to Willow Garage's ROS, for example.

The RB5X (shown in **Figure 6**) was an early personal robot first marketed in late 1982 by Joe Bosworth. The Androbot 'family'

shown in **Figure 7** (developed by Pong inventor, Nolan Bushnell at about the same time) was said to be the start of a new age of robotics. These bots were touted to be the first real personal robots for the average person's home. Both fell by the wayside, however, as it turned out the average consumer had no interest in programming, but simply wanted a robot that was ready to serve right out of the box. It was not until iRobot's Roomba hit store shelves that a robot worked right out of the box without complex programming and preparations, even though it's just a vacuum cleaner.

The Personal Assistant Robot

This is the type of robot that has long interested me as there are so many people who could use an assistant in their daily lives — especially the elderly. People with various disabilities require a companion that is designed around their particular disability or limitation. A paraplegic might have full function of their upper body and arms as well as their head and mouth, so a robot that can be voice or joystick-controlled for basic commands would be appropriate. A more severely disabled person might only have the ability to sip and puff through a special straw positioned near their mouth to form a command to their robot companion. There could be dozens of robot designs necessary to accommodate the many types of disabilities.

In comparison, the elderly typically have the physical movements and capabilities that a younger person does, but they are more limited and do not necessarily have the strength or dexterity they once did. Getting into and out of a bed, bathtub, or chair can be difficult if not impossible. The same applies for using a toilet.

Preparing meals requires reaching and lifting, and even simple washing at a sink can be difficult. The elderly can all benefit from the use of a personal assistant robot to extend their independence. These robots should be able to physically assist the person with strong, reliable, and functional arms, and not just be a rolling appliance with a mechanical voice to "remind Grandma to take her afternoon pill." This is the type of robot that I have been working on for the past 15 years.

FIGURE 6. Early RB5X ad.

FIGURE 7. Nolan Bushnell, Topo, and Bob.

Toyota Human Support Robot

If there's anything new in a full-size robot that is about to hit the world's market, there's a good chance that it's coming from Japan or Korea. Honda has long been known for its Asimo robot, but a newcomer in the field — Toyota — has brought forth the Human Support Robot (or HSR) shown in **Figure 8**. Designed to help the disabled to live independently, the HSR is Toyota's latest initiative in their Partner Robot program. It is intended to help around a person's home by retrieving hard-to-reach objects, picking things up from the floor,

FIGURE 9. Toyota HSR reach capability.

FIGURE 8. Toyota HSR Human Support Robot for the elderly.

FIGURE 10. Toyota HSR's head.

FIGURE 11. The RP-VITA from InTouch and iRobot.

FIGURE 12. iRobot AVA.

opening and closing curtains or windows, or just fetching something that the person needs but cannot get on their own. This robot is controlled via a tablet computer.

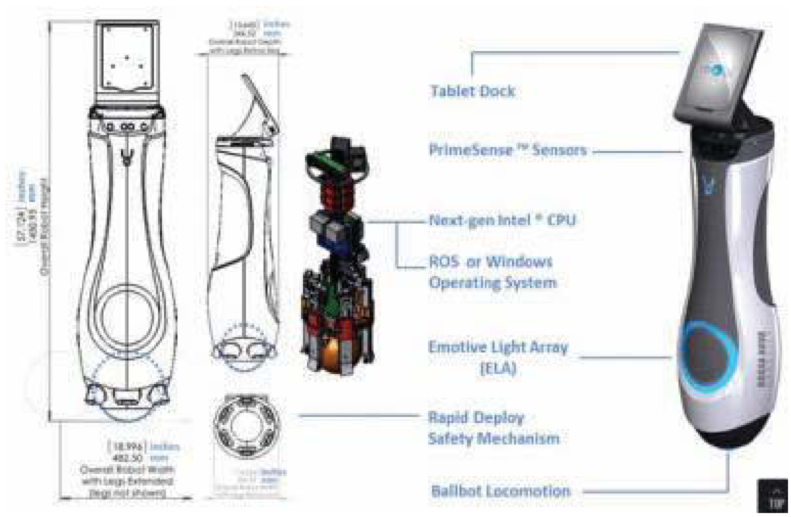
The unique part of the HSR that sets it apart from most home (and telepresence) robots is the articulated arm and telescopic body. The arm and two-fingered gripper can reach all the way to the floor — an important feature for a bed-ridden person. The telescopic body shown in **Figure 9** can extend the robot's height from 2.7 feet to 4.3 feet, and allow the 70 pound robot to manipulate objects up to a weight of 2.6 pounds. Its arm extends up to 2-1/2 feet. When the arm is not in use, it can fold into the robot's 14-1/2 inch diameter body.

Figure 10 shows a close-up view of the HSR's head, add-on tablet computer with built-in camera, separate cameras, and pattern recognition sensor. (Somehow, I feel that for the price of the final robot design, a built-in LCD display would be better than a tablet attached to the top of the robot.) The HSR has been undergoing extensive testing in cooperation with the Yokohama Rehabilitation Center since 2011, and actual patients have given valuable feedback.

InTouch Health and iRobot Debut RP-VITA

As I mentioned earlier, InTouch Health and iRobot have now joined forces to produce a remote presence robot (RP-VITA seen in **Figure 11**). Though this may seem like a bit of a setback from Toyota's HSR with a working arm, the RP-VITA is really more than just a telepresence robot. iRobot has already developed a hospital/corporate quality telepresence robot with the AVA robot shown in **Figure 12**.

The AVA is an autonomous development platform for commercial or personal robot projects. It features autonomous navigation and a sensor array consisting of laser, sonar, and 2D/3D imaging sensors. The 'head' can utilize standard LCD displays, tablets, or iPad-type display devices. Connectivity consists of Wi-Fi, Bluetooth, RF, and IR. Interfacing can be via touch screen, voice, gestures, and "person" following capabilities. The AVA utilizes an omni-directional drive for precise positioning and also



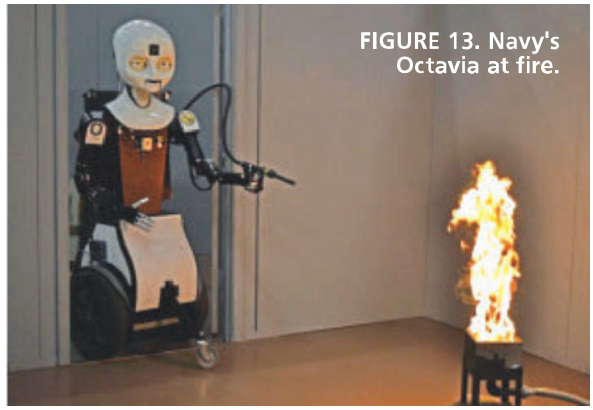


FIGURE 13. Navy's Octavia at fire.

FIGURE 14. mObi ball-bot from Bossa Nova Robotics.

has an adjustable height.

The modern hospital environment coupled with the dramatically rising expenses of the medical field demands a cost-effective method of placing a smaller number of medical doctors within a larger group of patients. Telepresence is proving to be the solution. That's one of the reasons iRobot and InTouch took the attributes of the original RP-7 (refer again to **Figure 1**) and the features of the AVA platform to create the resulting RP-VITA telemedicine platform.

As with the AVA platform, the RP-VITA has the same navigation capabilities in that it can be sent autonomously to a patient's room or follow a nurse or doctor on their rounds. Besides a touch-screen display that on-site medical professionals can use to call up a patient's history, the large LCD panel displays the face of the off-site doctor. Physicians can remotely access the robot via an iPad or similar device to obtain up-to-date information.

Tom Carroll can be reached at TWCarroll@aol.com.

Final Thoughts

Octavia (shown in **Figure 13**) was developed at the Navy's brand new 50,000 square foot Laboratory for Autonomous Systems Research to teach personnel how to work with robots, and for robots to learn, in turn, how to work with humans. Presently, Octavia is set up as a firefighter, since this can pose a serious problem on ships. The robot can determine the situation, use infrared cameras to sense the fire, and put it out with a special flame retardant. The media has happily called Octavia the new 'Rosie,' named after the Jetsons cartoon maid of the future. (Octavia actually reminds me of the revenge-seeking doll that tormented Telly Savalas in an old Twilight Zone episode.) This robot isn't something a family would have in their home, but it does help to highlight how so many organizations are developing helpmates for specific purposes.

Besides the few robots I selected for this article, there are some other amazing machines out there trying to catch the public's eye. The mObi from Bossa Nova Robotics (shown in **Figure 14**) is a single wheel — make that single *ball* — robot that is drawing the attention of the robotics world. It has extendable feet that quickly deploy to prevent it from tipping over when charging. The VGo and Double telepresence robots are among others that are making splashy headlines.

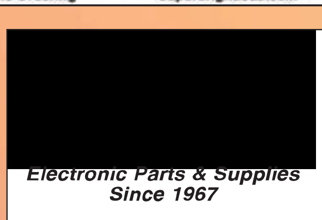

FIGURE 15. KUBI pan-tilt telepresence mount.

Finally, a telepresence platform that could work for probably all of us is the KUBI shown in **Figure 15** from Revolve Robotics based in San Francisco, CA. Tentatively priced at $249, the remotely-controlled pan and tilt mount holds an iPad, Surface, or other tablet with a camera, and communicates via its own app and Bluetooth 4.0.

These wildly different robots truly highlight what the inventive minds of today will develop for the homes of the future. **SV**

ROBO-LINKS

Got electronics?
www.nutsvolts.com



For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links.

Advertiser Index

DYNAMIXEL PRO

The Solution for Full Scaled Robots

High Power, High Precision

Up to 200W BLDC motor and up to 500,000
pulse per turn (4096 absolute resolution)

Full Modular Solution

Ease of implementation (no additional design for parts needed)
Ease of maintenance (simple part swapping)

Sophisticated Control Algorithms

Position and speed input commands
with dual-loop current control

Novel Gear Reduction System

High torque output, light weight,
with high impact resistance

Data		H42-20-S300-R		H54-100-S500-R		H54-200-S500-R	
		Unit	Data	Unit	Data	Unit	Data
Rated voltage		V	24	V	24	V	24
No load speed		RPM	28.3	RPM	35.2	RPM	35
No load current		A	0.61	A	1.06	A	1.18
Continuous operation	Speed	RPM	15.59	RPM	32.7	RPM	32.1
	Torque	Nm	5.596	Nm	21.142	Nm	39.131
Maximum output power		W	23.64	W	144.58	W	262.66
Resolution		Step/turn	304,000	Step/turn	502,000	Step/turn	502,000
Gear ratio		-	304	-	502	-	502
Backlash		arcmin	3.5	arcmin	3.5	arcmin	3.8
Interface		-	RS-485 / CAN	-	RS-485 / CAN	-	RS-485 / CAN
Operating temperature		˚C	5~55	˚C	5~55	˚C	5~55


DARwIn-OP

ROBOTIS
www.robotis.com

USA E-mail : america@robotis.com, Tel : +1-949-333-3635
JAPAN E-mail : japan@robotis.com, Tel : +81-3-4330-3660
Etc. E-mail : contactus2@robotis.com, Tel : +82-70-8671-2609

Life with Robot!
Not a dream anymore

HerkuleX,
The Best Investment for your robot project.

Powered by
FAULHABER

Dongbu Robot is very proud to introduce our new high-end DRS-040X / 060X servo series.

- ✓ Superior Resolution / Monster Torque
- ✓ Excellent Linearity
- ✓ Durable Metal Gear Train
- ✓ Premium FAULHABER motor inside
- ✓ Compact Design
- ✓ Incremental Magneto Resistant Encoder (For DRS-0402 / 0602)

HerkuleX Smart Servo Series

	DRS-0601	DRS-0602	DRS-0401	DRS-0402	DRS-0201	DRS-0101
Size	56mm(W) X 35mm(D) X 46mm(H)		56mm(W) X 35mm(D) X 38mm(H)		45mm(W) X 24mm(D) X 32mm(H)	
Weight	145g		123g		60g	45g
Input Voltage	9.5 ~ 14.8V DC				7 ~ 12V DC	7.4V DC
Stall Torque	77.0kgf.cm@14.8V / 62.5kgf.cm@12.0V		52.0kgf.cm@14.8V / 43.1kgf.cm@12.0V		24.0kgf.cm@7.4V	12.0kgf.cm@7.4V
Max. Speed	0.164s/60°@14.8V		0.162s/60°@14.8V		0.147s/60°@7.4V	0.166s/60°@7.4V
Operating Angle	320°, Continuous Rotation	720°, Continuous Rotation (Built-in Speed Controller)	320°, Continuous Rotation	720°, Continuous Rotation (Built-in Speed Controller)	320°, Continuous Rotation	
Resolution	2048 Steps	12960 Steps	2048 Steps	12960 Steps	1024 Steps	
Communication	Full Duplex Asynchronus Serial(TTL), Multi Drop, 0~254 ID, Max 1Mbps (0.67Mbps for DRS-0101, DRS-0201)					
Motor	Premium FAULHABER Coreless DC				Coreless DC	Metal Brush Cored DC
Position Sensor	Potentiometer	Magnetic Encoder	Potentiometer	Magnetic Encoder	Potentiometer	
Gear	Super Reinforced Metal					Engineering Plastic
Feedback	4 Position, Speed, Temperature, Overload, Voltage, Real Time Tick, etc.					
Features	PID, Feedforward, Trapezoidal Velocity Profile, Velocity Override, Torque Saturator & Offset, Overload Prevention, Absolute Position Adjustment, Deadband, etc.					
Programming	More than 50 parameters settings available via HerkuleX Manager Kit					

Dongbu Robot Co., Ltd. www.dongburobot.com

Headquarter : 11th Floor, Bucheon Techno Park 401, Yakdae-dong, Wonmi-gu, Bucheon-city, Gyunggi-do 420-734, Korea
TEL : +82-32-329-5551(ext.311), FAX : +82-32-329-5569, E-MAIL : robotsales@dongbu.com
Factory : 27, 6 Gil, 4 Sandan, Jiksan-eup, Seobuk-gu, Cheonan-city, Chungcheongnam-do 331-814, Korea, TEL : +82-41-590-1700, FAX : +82-41-590-1701

Dongbu Robot